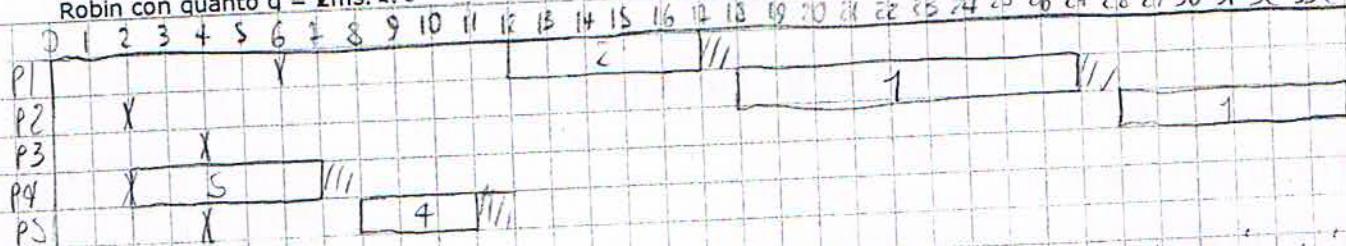


4. $3 \times 2 = 6$ punti (PER CFU 10 e 12)

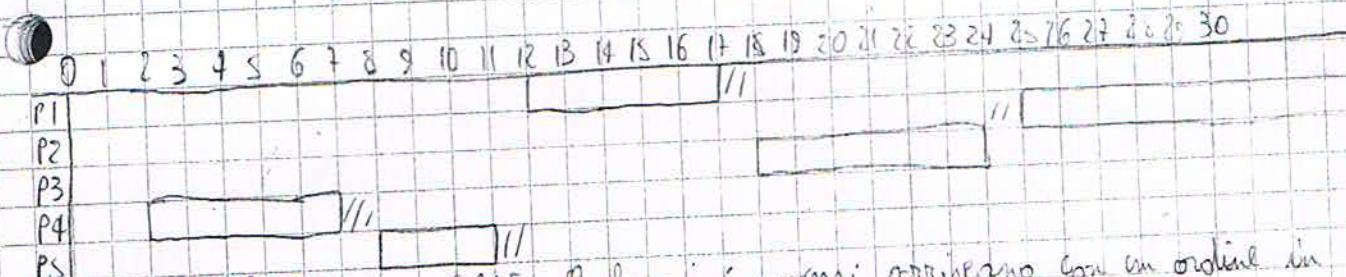
Si considerino i seguenti processi, attivi in un sistema multibordo:

Processo	Tempo di arrivo	CPU-burst	Priorità
P1	6ms	5ms	2
P2	2ms	9ms	1
P3	4ms	6ms	1
P4	2ms	5ms	5
P5	4ms	3ms	4

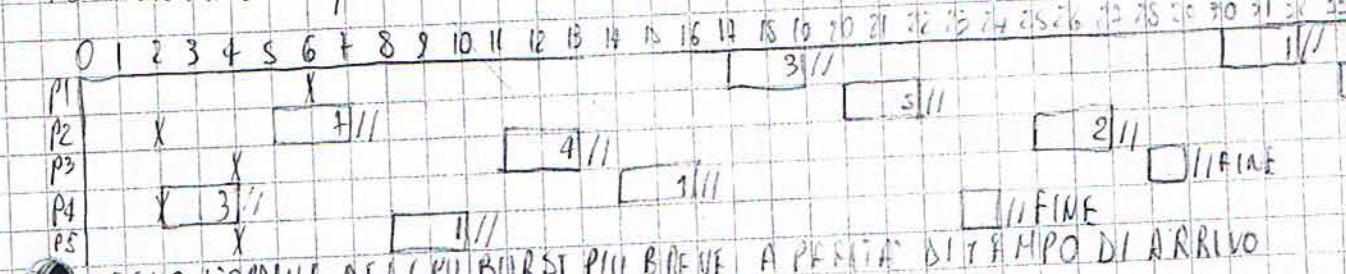
Supponendo che il cambio di contesto sia 1ms, si mostri l'ordine di esecuzione dei processi e quanto vale il tempo di attesa medio, il tempo di turnaround medio ed il tempo di turnaround normalizzato medio per ciascuno dei seguenti algoritmi di scheduling della CPU: (a) Priorità con prelazione (la priorità massima è 5) e (b) Round-Robin con quanto $q = 2$ ms.



PRIORITÀ CON PRELAZIONE Tutti i processi arrivano in un ordine in cui non si disturbano a vicenda e riescano ad eseguire per intero quello a priorità maggiore. Per i processi P2 e P3 che hanno stessa priorità, le dàlo priorità a quello arrivato prima.

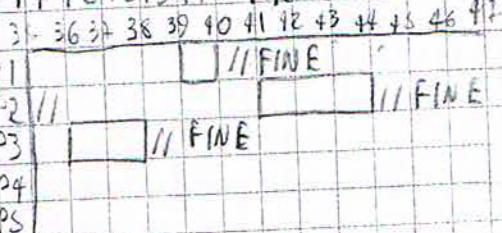


SJF CON PRELAZIONE => SJTF Anche qui i processi arrivano con un ordine in cui non dobbiamo prelazionare.

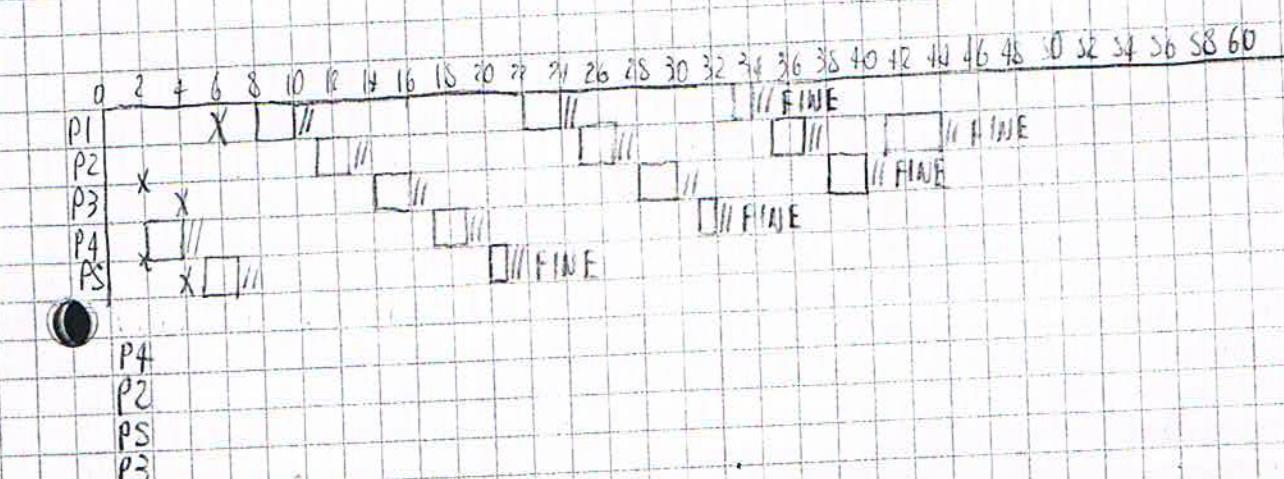


SEGUO L'ORDINE NEI CPU BURST PIÙ BREVI A PRIORITÀ DI TEMPO DI ARRIVO

CODA: P4 P2 P3 P1 - RR NORMALE



OPPURE P4 dato che P4 si rivolte dopo P2 all'istante 4 dopo P2-P1-P3, P5 più tardi che arriverà P1.
P3
P4
P1
P2 P3 P1 P2 P4



TEMPO DI ATTESA MEDIO: $(6+16+24+10)/4 = 15$ THROUGHPUT: $\frac{S}{3+UNITA'DI+TEMPO}$

TEMPO TURNAROUND MEDIO: $C1+C2+C3+C4 = 5+9+6+5 = 25$

TEMPO TURNAROUND NORMALIZZATO: $P1 = \frac{11}{5}, P2 = \frac{25}{9}, P3 = \frac{30}{6}, P4 = \frac{5}{5} \cdot PS = \frac{7}{4}$
al determinare abbiamo il Tempo di servizio, ovvero il Tempo per il quale un processo che possiede la risorsa CPU.

SJF CON PRELAZIONE
TEMPO DI ATTESA MEDIO

TEMPO TURNAROUND MEDIO:

TEMPO TURNAROUND NORMALIZZATO:

ROUND ROBIN NORMALE

TEMPO DI ATTESA MEDIO.

TEMPO TURNAROUND MEDIO.

TEMPO TURNAROUND NORMALIZZATO.

ROUND ROBIN CON PRIORITA'

TEMPO DI ATTESA MEDIO: $(23+31+30+25+14)/5$

TEMPO TURNAROUND MEDIO.

TEMPO TURNAROUND NORMALIZZATO.

Si supponga che in un sistema siano presenti processi P0, P1, P2, P3, P4 ed un insieme di risorse di quattro tipi diversi A, B, C e D e di trovarsi nella seguente configurazione:

Risorse allocate	Risorse massime	Risorse disponibili						
A	B	C	D	A	B	C	D	
P0	6	0	Z+4	2	8	4	5	6
P1	10	0	0	2	12	7	6	8
P2	6	2-X	0	0	8	2	0	8
P3	0	0	3	2	0	3	4	2
P4	4	1	2-Y	4	11	1	6	9

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro.

NEED: A B C D

$$P0 \quad 2 \quad 4 \quad 1-Z \quad 4$$

$$P1 \quad 2 \quad 7 \quad 6 \quad 6$$

$$P2 \quad 2 \quad X \quad 0 \quad 8$$

$$P3 \quad 0 \quad 3 \quad 1 \quad 0$$

$$P4 \quad 1 \quad 0 \quad Y+1 \quad 8$$

PARTE P0:

$$\begin{cases} 1-Z \geq 10 \\ 1-Z \geq 0 \end{cases} \quad \begin{cases} 4 \geq 9 \\ 4 \geq -1 \end{cases} \quad \begin{cases} 2 \geq 9 \\ 2 \geq 1 \end{cases} \quad \begin{array}{|c|c|c|c|c|c|} \hline & & & -4 & 0 & +1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|c|c|c|} \hline & & & -9 & 5 & 2 \geq 1 \\ \hline \end{array}$$

DISP: 12 4 2+14 6

PARTE P3: (l'intervalle delle risorse C (-9+14, 1+14) sono maggiori di "1")

DISP: 12 (4) 2+14 8

PARTE P2:

$$\begin{cases} X \geq 0 \\ X \leq 4 \end{cases} \quad \begin{array}{|c|c|c|c|c|c|} \hline & 0 & 4 & & & \\ \hline \end{array} \quad 0 \leq X \leq 4$$

DISP 18 6-X 2+14 8

PARTE P4

$$\begin{cases} Y+4 \geq 0 \\ Y+4 \geq 17 \end{cases} \quad \begin{cases} Y \geq -4 \\ Y \geq -17+2 \end{cases} \quad \begin{cases} Y \geq -4 \\ Y \geq 13+2 \end{cases} \quad \begin{array}{|c|c|c|c|c|c|} \hline & -4 & 0 & 13+2 & & \\ \hline \end{array}$$

DISP 22 (7-X) 19+Z-Y 12

PARTE P1:

$$\begin{cases} 7 \leq 7-X \\ 7 \leq 0 \end{cases} \quad \text{RESTRIGO L'INTERVALLO DA } 0 \leq X \leq 7 \text{ A } X=0$$

$$\begin{cases} 6 \leq 19+Z-Y \\ 6 \leq 19 \end{cases} \quad \begin{cases} -Z+Y \leq -6+19 \\ -Z+Y \leq 13 \end{cases} \quad \begin{cases} Z-Y \geq -13 \\ Z-Y \leq 13 \end{cases} \quad \text{dato che il valore massimo è} \\ Z=1, e Y \leq 13+1, \quad 1-1 \leq -13 \quad -13 \leq -13$$

Quindi entro questo intervallo il processo

Ogniore ristrikendo solo in altro attica l'intervallo massimo:

$$6 \leq 19+1-14 \quad 6 \leq 6$$

1. $3 \times 3 = 9$ punti

Si consideri la seguente tabella delle pagine per un processo con 10 pagine virtuali, 6 frame, dimensione della pagina 1024 parole

Pagina	Tempo di caricamento	Tempo ultimo di riferimento	Bit modifica	Bit riferimento
0	50	80	1	0
1	10	130	0	0
2	60	70	1	0
3	-	-	0	0
4	90	200	1	1
5	-	-	0	0
6	30	85	1	0
7	55	75	1	1
8	200	250	0	1
9	20	300	0	1

Determinare per la stringa di riferimento:

0 4 5 9 7 8 3 9 5 0 6 4 1 3 2 7

le pagine che vengono sostituite per gli algoritmi (a) FIFO (b) LRU e (c) CLOCK, supponendo che l'ultima pagina cambiata con questo algoritmo sia stata la pagina contenuta nel quarto frame. Si determini, inoltre, per ognuno dei seguenti indirizzi virtuali, l'indirizzo fisico generato corrispondente: 2820 825 8034 4100 10000

FIFO	0	4	5	9	7	8	3	9	5	0	6	4	1	3	2	7
CS	0	8	5	9	9	9	3	3	3	0	6	4	1	1	2	7
1	4	4	8	5	5	5	9	9	9	3	0	6	4	4	1	2
2	2	2	8	4	8	3	8	3	8	3	0	6	6	4	1	
3	7	1	7	6	4	4	8	8	8	3	9	3	0	0	6	4
4	0	0	0	3	2	2	2	4	4	4	8	3	9	3	3	0
S	6	6	6	0	7	7	2	2	2	4	8	3	9	3	0	
	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	

HO ASSUNTO DI TROVARMI AL MOMENTO 201

Carico le pagine dal + grande al + piccolo
Sposto in testa alla pila la pagina di stringa.
dei gestiti (E non testa)

LRU	0	4	5	9	7	8	3	9	5	0	6	4	1	3	2	7
0	9	0	4	5	9	7	8	3	9	5	0	6	4	1	3	2
1	8	9	0	4	5	9	7	8	3	9	5	0	6	4	1	3
2	4	8	9	0	4	5	9	7	8	3	9	5	0	6	4	1
3	1	4	8	9	0	4	5	9	7	8	3	9	5	0	6	4
4	6	1	1	8	8	0	4	5	9	7	8	3	9	5	0	6
S	0	6	6	1	1	8	0	4	4	4	7	8	3	9	5	0
	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	

Carico le pagine dal + grande al più piccolo
dell'orario di riferimento.
Inserito in testa alla pila TUTTE le pagine
della stringa.

CLOCK	0	4	5	9	7	8	3	9	5	0	6	4	1	3	2	7
0	8	x	8	x	8	x	8	x	8	x	8	x	x	x	x	x
1	4	x	4	x	4	x	4	x	4	x	0	0	0	0	x	x
2	2	2	2	9	x	9	x	9	x	9	x	9	x	x	x	x
3	F	x	7	x	7	x	7	x	7	x	7	x	x	x	x	x
4	0	x	0	x	0	x	0	x	0	x	3	x	x	x	x	x
S	6	6	6	S	x	S	x	S	x	S	x	S	x	x	x	x
	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	

L'ordinamento è per tempo di caricamento
con bit di riferimento.
è ultima pagina inserita in al 3° frame,
quindi si ottiene il bit di ref.

er gli indirizzi da virtuali a fisici:

2820 = 2820 1024 quanti pagine 2 offset 772 [2] [7] [7] [2] prendendo l'algoritmo

FIFO come riferimento, al momento del primo caricamento delle pagine (8+4+2+0+6) la pagina 2 è localizzata nel 3° frame, ovvero il 3 (perché si parte da 0) quindi diventa [2] [7] [7].

C'è un caso che sia uguale al logico) se la pag logica non è presente in memoria,

la carica nel primo frame e l'indirizzo sarà [0] [OFFSET].

EAT

TEMPO ACCESSO TLB = 50 ns

TEMPO ACCESSO MEMORIA = 150 ns

TEMPO X GESTIONE PF = 8 ns \Rightarrow 8 000 000 ns

PAGE FAULT RATE 6% $0,06 \Rightarrow$ MEMORIA HIT RATE = 94% 0,94

$$TLB \text{ HIT RATE } 60\% = 0,6$$

FORMULA GENERALE

$$PR_2(T_{TLB} + T_{MEM}) + (PA_1 - PR_2)(T_{TLB} + 2T_{MEM}) + (1 - PR_1)(T_{TLB} + T_{MEM} + T_{PF} + T_{SWAP})$$

dove: PR_1 è la possibilità di trovare la pagina in memoria
 PR_2 è la

$1 - PR_1$ è la non trovare la pagina in memoria nel TLB
 TLB, e quindi viene generato PF

$$0,6(50 + 150) + (0,94 - 0,60)(150 + 300) + 0,06(50 + 150 + 8000000 + 50 + 300)$$

$$0,6(200) + (0,34)(450) + 0,06(800050)$$

$$120 + 119 + 480033 = EAT \text{ ns}$$

b) TLB MISS RATIO = 20%

IL 5% NEGLI INDIRIZZI LOGICI GENERA PF

NEL 60% DEI CASI DI PF LA PAGINA VITIMA HA DIRTY BIT = 0

TEMPO DI SWAP = 3 ms

$$PR_2 = 1 - 20\% = 80\%$$

$$PR_1 = 1 - 5\% = 95\%$$

// nel caso di dirty bit uguale a 0 ms deve considerare SWAP TIME

$$0,8(50 + 150) + (0,95 - 0,80)(50 + 300) + (0,05 \times 0,60)(50 + 150 + 8000000 + 50 + 300)$$

$$+ (0,05 \times 0,40)(50 + 150 + 8000000 + 30000000 + 50 + 300)$$

LA FORMULA FINALE UTILIZZATA E' SARA':

$$PR_2(T_{TLB} + T_{MEM}) + (PA_1 - PR_2)(T_{TLB} + T_{MEM}) + (PF_RATE \times DIRTY_BIT_0)(T_{TLB} + T_{MEM} + PF + T_{SWAP} + T_{TLB} + 2T_{MEM})$$

$$+ (PF_RATE \times DIRTY_BIT_1)(T_{TLB} + T_{MEM} + PF + T_{SWAP} + T_{TLB} + 2T_{MEM})$$

FORMULA CON DIRTY BIT

4 GIUGNO 2010

21

$$2 \cdot 3^3 = 9 \text{ punti}$$

Un disco ha 7000 cilindri, 200 tracce per cilindro, 100 settori per traccia, dimensione del settore pari a 1024B, tempo di seek di 10 ms per ogni cilindro attraversato, e un tempo di attraversamento di tutti i cilindri pari a 3 ms, frequenza di rivoluzione pari a 5400 rpm. Inoltre si fanno le seguenti ipotesi: la testina è inizialmente posizionata sul cilindro 140; l'attesa prima che il settore desiderato passi sotto la testina dopo il seek è di una rotazione; il disco dispone di un buffer sufficiente per memorizzare un solo settore alla volta. Calcolare il tempo impiegato a soddisfare le seguenti richieste* di 4 settori medi ciascuno:

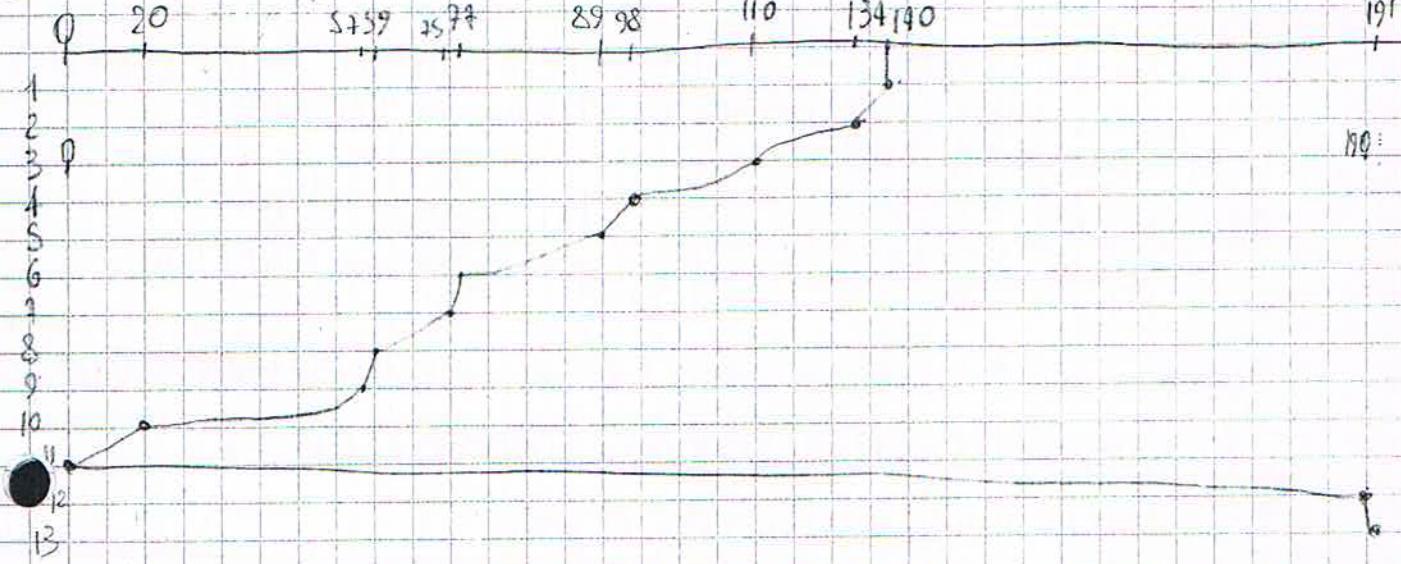
134, 140, 1391, 477, 1489, 1059, 1220, 1075, 1110, 57, 98 secondo le politiche: a) SCAN, b) C-LOOK e c) SSTF nel caso che arrivino al tempo 60 ms la richiesta di lettura di 6 settori per la traccia 590 e all'istante 120 ms la richiesta per la traccia 1000 di 5 settori.

7000 CILINDRI, 200 TRACCE PER CILINDRO, 100 SETTORI PER TRACCIA, BLOCCO 1024B
10ms SEEKTIME ATTRaversamento 3ms, 5400 RPM, POSIZIONE TESTINA 140,
ROTAZIONE N'TANDO ROTAZIONALE, 4 SETTORI IN CHIESA

* I BANDVIZIONI DABBI OCCORRE A TRACCE:

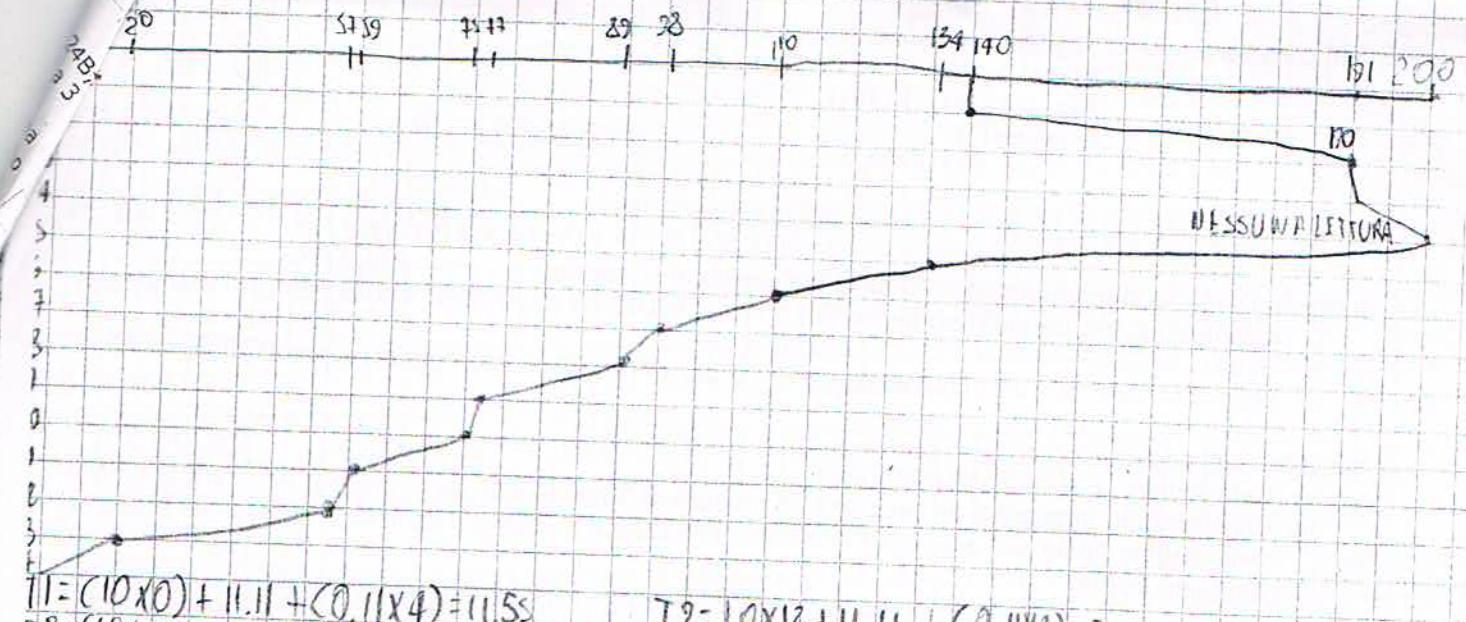
$134 \% 200 = 134$	$1489 \% 200 = 89$	$1110 \% 200 = 110$
$140 \% 200 = 140$	$1059 \% 200 = 59$	$57 \% 200 = 57$
$1391 \% 200 = 91$	$1220 \% 200 = 20$	$98 \% 200 = 98$
$477 \% 200 = 77$	$1075 \% 200 = 75$	

SSTF $5400 \text{ RPM} \Rightarrow \frac{5400}{60} = 90 \text{ RPS}$ RETRASO = $\frac{1}{90} = 11.1 \text{ ms}$ SINGOLO SETTORE = $0.11 / 100 =$



$$\begin{aligned}
 T_1 &= (10 \times 0) + 11.11 + (0.11 \times 4) = 11.11 + (10 \times 16) + 11.11 + (0.11 \times 4) = 171.55 \\
 T_2 &= (10 \times 6) + 11.11 + (0.11 \times 4) = 71.55 + 11.11 + (0.11 \times 2) = 81.55 \\
 T_3 &= (10 \times 24) + 11.11 + (0.11 \times 4) = 251.55 + 11.11 + (0.11 \times 5) = 281.55 \\
 T_4 &= (10 \times 12) + 11.11 + (0.11 \times 4) = 131.55 + 11.11 + (0.11 \times 20) = 161.55 \\
 T_5 &= (10 \times 9) + 11.11 + (0.11 \times 4) = 101.55 + 11.11 + (0.11 \times 190) = 111.55 \\
 T_6 &= (10 \times 1) + 11.11 + (0.11 \times 4) = 11.55 + 11.11 + (0.11 \times 1) = 21.55 \\
 T_7 &= 10 \times 2 = 20
 \end{aligned}$$

* L'esercizio è stato interpretato male in base alla Prossia, ma comunque lo sviluppo è fatto bene. L'esercizio è stato imposto in modo che ogni lavorazione si trovi all'ultima Prossia, dove esistono 200 tracce su un piatto (e non 7000).



$$T_1 = (10 \times 0) + 11.11 + (0.11 \times 4) = 11.55$$

$$T_2 = (10 \times 5) + 11.11 + (0.11 \times 6) = 511.77$$

$$T_3 = 10 \times 1 + 11.11 + (0.11 \times 4) = 21.55$$

$$T_4 = 10 \times 9 = 90$$

$$T_5 = 10 \times 66 + 11.11 + (0.11 \times 4) = 671.55$$

$$T_6 = 10 \times 24 + 11.11 + (0.11 \times 4) = 251.55$$

$$T_7 = 10 \times 12 + 11.11 + (0.11 \times 4) =$$

$$T_8 = 10 \times 9 + 11.11 + (0.11 \times 4) =$$

$$T_9 = 10 \times 12 + 11.11 + (0.11 \times 4) =$$

$$T_{10} = 10 \times 2 + 11.11 +$$

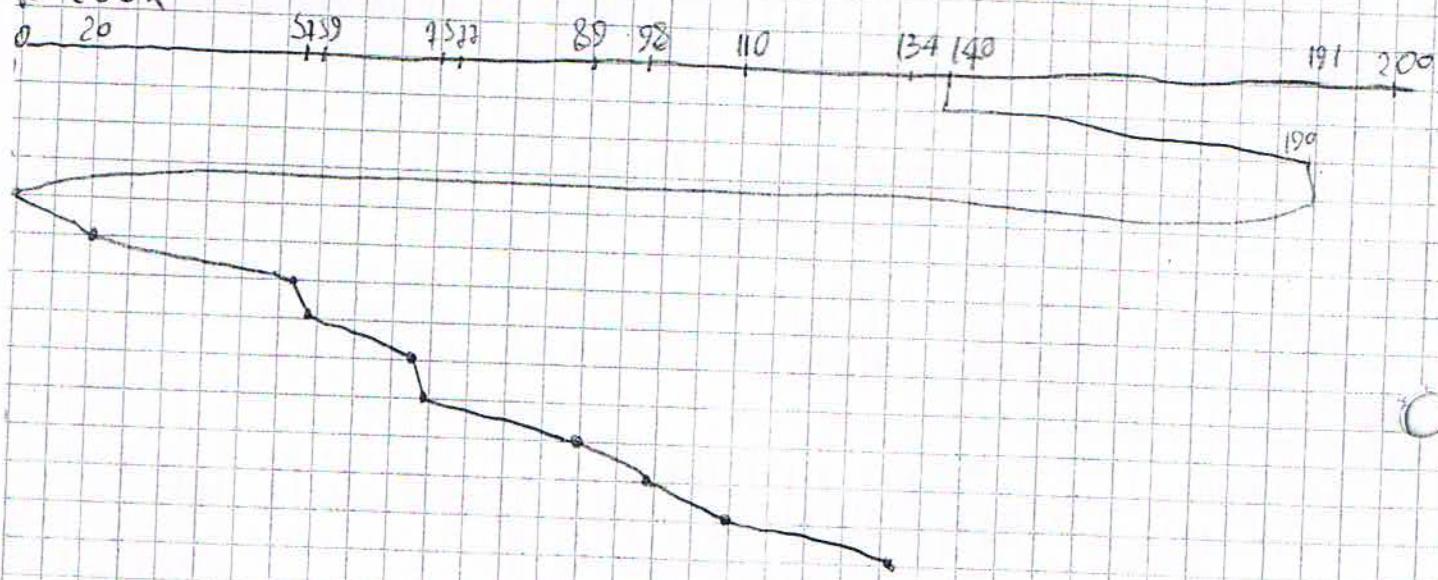
$$T_{11} = 10 \times 16 + 11.11 +$$

$$T_{12} = 10 \times 2 + 11.11 +$$

$$T_{13} = 10 \times 32 + 11.11 +$$

$$T_{14} = 10 \times 20 + 11.11 + (0.11 \times 5) =$$

C-LOOK



$$T_1 = 10 \times 0 + 11.11 + 0.11 \times 4 = 11.55$$

$$T_2 = 10 \times 5 + 11.11 + 0.11 \times 6 =$$

$$T_3 = 10 \times 1 + 11.11 + 0.11 \times 4 =$$

$$T_4 = 3 \text{ ms (A17R)} + 10 \times 0 + 11.11 + 0.11 \times 5 =$$

$$T_5 = 10 \times 20 + 11.11 + 0.11 \times 5 =$$

$$T_6 = 10 \times 32 + 11.11 + 0.11 \times 4 =$$

$$T_7 = 10 \times 2 + 11.11 + 0.11 \times 4 =$$

$$T_8 = 10 \times 16$$

$$T_9 = 10 \times 2 + 11.11 + 0.11 \times 4 =$$

$$T_{10} = 10 \times 12$$

$$T_{11} = 10 \times 9$$

$$T_{12} = 10 \times 12$$

$$T_{13} = 10 \times 24$$

1bis. 3*3= 9 punti

Prendiamo in considerazione la seguente porzione di codice, con ogni int che occupa 8B:

```
#define N 1024
int a[4][N], b[4N];
compute(){
    unsigned i, j;
    for (i=0; i<4; i++)
        for (j=0; j<N/2; j++)
            a[i][2*j/3] = b[4*j] + a[i][j+N/3];}
```

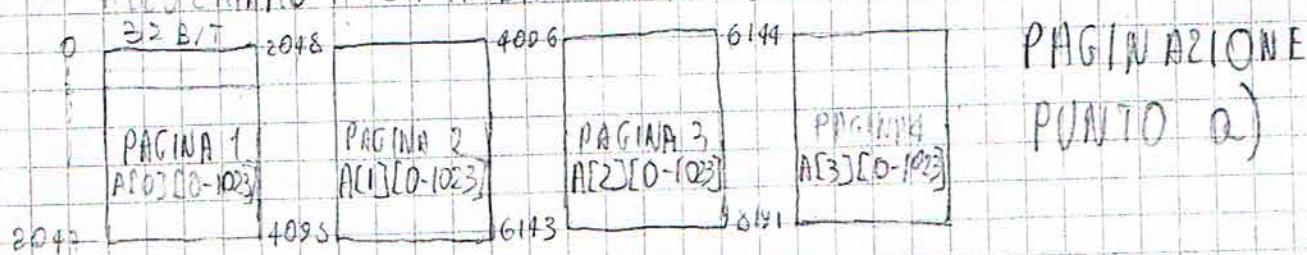
- 1) Si indichi lo spazio di memoria allocato nelle due seguenti situazioni: PAGINA
- paginazione con un indirizzo logico di 20 bit, di cui 11 bit per l'offset e 9, e parole di memoria di 32bit.
 - segmentazione con un indirizzo logico di 16 bit, di cui 10 bit per l'offset, e parole di memoria di 32bit.
- 2) Nel caso della paginazione:
- Determinare la stringa dei riferimenti
 - Considerando di avere a disposizione un numero di 3 frame dati per processo, mostrare la tabella delle pagine finale con algoritmo di sostituzione CLOCK e ultima pagina sostituita quella presente nel terzo frame.

$$A = [4][1024]; B = [4][1024] \Rightarrow A = 1024 \times 4 \times 8 \text{ BYTES} = 32768; B \text{ uguale a } 32768$$

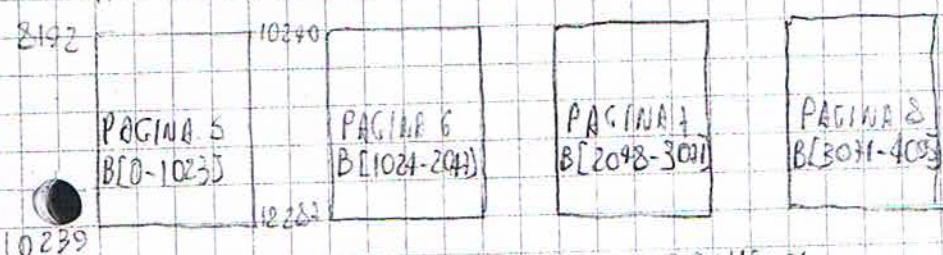
LA PAGINA HA UNA DIMENSIONE (OFFSET) di $2^9 = 512 \text{ BYTES}$

Ogni parola è 32 BIT 4 BYTES. QUINDI OGNI RIGA DELLA TABELLA (DITTELE 14 BYTES)

ALLOCHIAMO A: C'È UNO OCCUPA QUINDI 2 ALIGHI, NECESSARI 4096 INTERI



ALLOCHIAMO B. NECESSARI 4096 INTERI (IDENTICO AL A)



DETERMINA LA STRINGA DEI RIFERIMENTI
 $A[0][340] = B[2044] + A[0][852] \Rightarrow 1, S, 1$, dopo mossa for: 1, S, 1

$A[1][340] = B[2044] + A[1][852] \Rightarrow 2, S, 2$ 2, S, 2

$A[2][340] = B[2044] + A[2][852] \Rightarrow 3, S, 3$ 3, S, 3

$A[3][340] = B[2044] + A[3][852] \Rightarrow 4, S, 4$ 4, S, 4

Quindi 1-S-1 per le prime mose del ciclo, poi tutte le seconde mose

2-S-2

3-S-3

4-S-4

4-S-4

Quindi 1-S-1-6-1-2-S-2-6-2|3-S-3-6-3|4-S-4-6-4

INTRODUZIONE

N10 B)

Ogni segmento è $2^10 = 1024$ BYTES

ALLOCAZIONE MATRICE A: $A[4][1024]$ VETTORE B

$A[0][0-511]$	$A[0][512-1023]$	$A[1][0-511]$	$A[1][512-1023]$	$A[2][0-511]$	$A[2][512-1023]$
1	2	3	4	5	6
$A[3][0-511]$	$A[3][512-1023]$	$B[0-511]$	$B[512-1023]$	$B[1024-1535]$	$B[1536-2047]$
7	8	9	10	11	12
$B[2048-2559]$	$B[2560-3183]$	$B[3184-3695]$	$B[3696-4096]$		
13	14	15	16		

HO QUINDI 16 SEGMENTI

3. $2 \times 3 = 6$ punti

Si consideri la gestione della memoria a partizioni variabili con politica di assegnazione delle partizioni *worst fit*. si prenda in esame un processo A che occupa le partizioni di origine 21300, 40500 e 62000 e lunghezze 10000, 7000 e 2000 rispettivamente per i segmenti codice, dati e pila. Ad un certo istante il processo A esegue la chiamata fork che genera il processo B. Al momento della chiamata sono libere 3 partizioni, che iniziano rispettivamente agli indirizzi 77000, 90000 e 400000 e hanno rispettivamente lunghezze 12000, 10000, e 20000. Indicare

- l'origine e la lunghezza delle partizioni libere dopo l'esecuzione della fork ;
- i contenuti dei registri base e limite dei segmenti del processo A durante la sua esecuzione
- i contenuti dei registri base e limite dei segmenti del processo B durante la sua esecuzione.

WORST FIT: Allocazione nello spazio di memoria + grande.

PROCESSO A IN FASE DI PARTENZA

	21300	40500	62000	77000	89000	100000	
CO	DA	P		FREE	FREE		FREE
DI	TI	LA					
CE							

21300 40500 62000 77000 89000 100000
400000

PROCESSI A & B DOPO LA FORK (ho copiato nell'ordine CODICE, DATI, PILA)

	CODI	DA	PI	DATI	FI	PI	
CE A	II	LA	B	R	LA		CODICE
	A	A	F	F	E		B
			77000	89000	90000		400000

HO ASSUNTO CHE SUPRANIE QUESTA FORK NON E' VISTA IL COPY ON WRITE

HO ASSUNTO CHE LA FORK COPIA PRIMA L'AREA CODICE, Poi DATI E Poi PILA

✓ ✓ ✓ ✓ - A PARITA DI SPAZIO DISPONIBILE 10000 (40000-420000=90000) LA SCELTA IN CUI COLLOCARE LA PILA DI B SIA ARBITRARIA.

6 punti (PER CFU 10 e 12)

Si consideri un file system che utilizza 16 blocchi fisici ed usa la tecnica della bitmap per tenere traccia della memoria libera. Si consideri la seguente configurazione iniziale della bitmap:

0000 1100 1001 1010

con la convenzione che nella bitmap lo stato del blocco 0 è rappresentato dal bit più a sinistra e 1 rappresenta i blocchi occupati. Supponendo che il file system allochi sempre i blocchi di indice minore, descrivere la struttura della bitmap dopo le seguenti operazioni:

- a) scrittura del file A (4 blocchi); b) scrittura del file B (3 blocchi); c) aggiunta di due blocchi ad A cancellazione di B

"DAL BIT PIÙ A SINISTRA" indica che l'allocazione dei blocchi avviene da sinistra a destra

SCRITTURA FILE A (4 BLOCCHI)

||||| 1100 1001 1010
A A A A

SCRITTURA FILE B (3 BLOCCHI)

||||| 1111 1101 1010
A A A A B B B

AGGIUNGERE 2 BLOCCHI AD A E CANCELLARE B

||||| 1100 1011 1110
A A A A A A

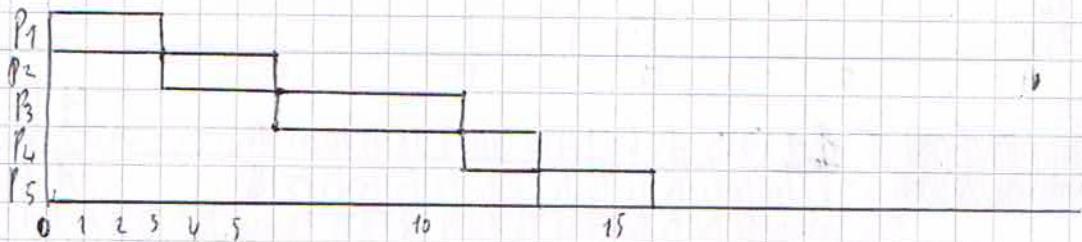
Ho utilizzato la politica concatenata, dato che con la singola non riesco a spazio e non posso scrivere l'indice minore, cioè la condizione di poter allocare blocchi dove voglio, dato che sono vincolato all'allocare blocchi di indice minore come indica il testo.

SCHEDULING NON-PREEMPTIVE

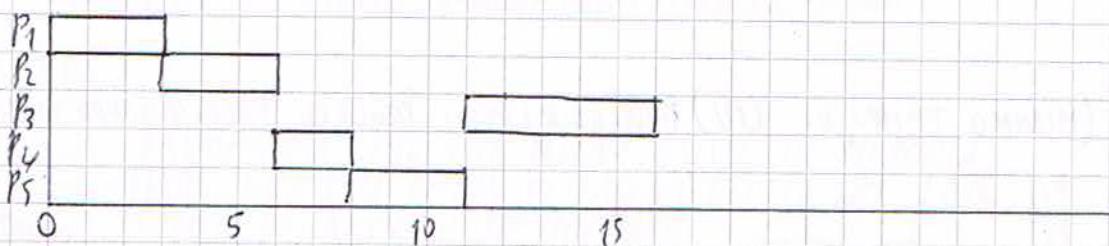
- FCFS (SCHEDULATI COSÌ COME ARRIVANO)
- SJF (SCHEDULA RICHIESTA CON MINIMO TEMPO DI SERVIZIO)
- HRN (SCHEDULA IL PROCESSO CON IL PIÙ ELEVATO RAPPORTO DI RISPOSTA)

PROCESSO	P ₁	P ₂	P ₃	P ₄	P ₅
TEMPO AMMISSIONE	0	2	3	4	8
TEMPO DI SERVIZIO	3	3	5	2	3

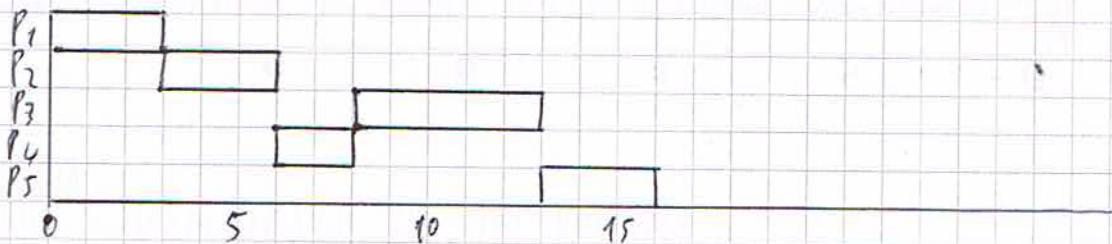
FCFS



SJF



HRN



RAPPORTO DI RISPOSTA: TEMPO DI ATTESA + TEMPO DI SERVIZIO / TEMPO SERVIZIO PROCESSO

TEMPO SERVIZIO / TEMPO SERVIZIO PROCESSO

$$0 \text{ } P_1 \\ 3 \text{ } (P_2 = 3 - 2^3/3 = 1,33) \\ 6 \text{ } (P_3 = 6 - 3 + 5/5 = 1,6) \\ 8 \text{ } (P_4 = 8 - 3 + 5/5 = 2) \\ 16 \text{ } (P_5 = 16 - 8 + 3/3 = 1)$$

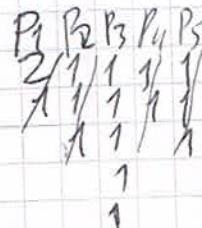
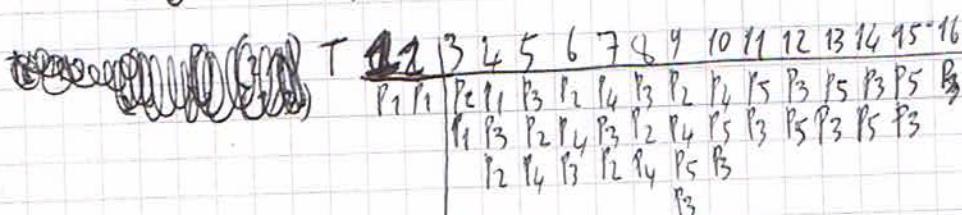
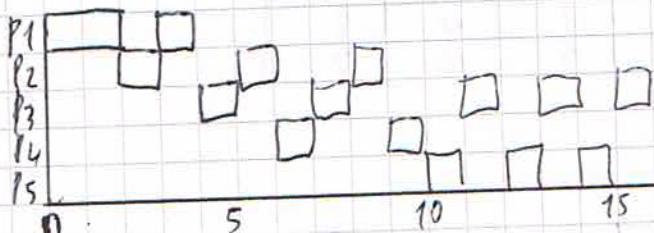
COME LI ABBIAMO CALCOLATI?

$$\frac{(TEMPO SCHEDULATO - TEMPO ARRIVO + TEMPO SERVIZIO)}{TEMPO SERVIZIO}$$

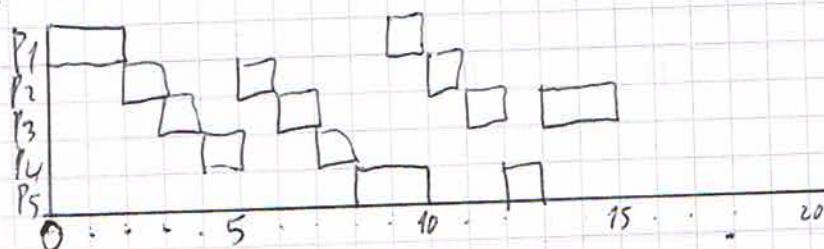
- SCHEDULER PREEMPTIVE**
- ROUND ROBIN (RR)
 - LEAST COMPLETED NEXT (LCN)
 - SHORTEST TIME TO GO (STG)

PROCESSO	P1	P2	P3	P4	P5
TEMPO AMM.	0	2	3	4	8
TEMPO SERV.	3	3	5	2	3

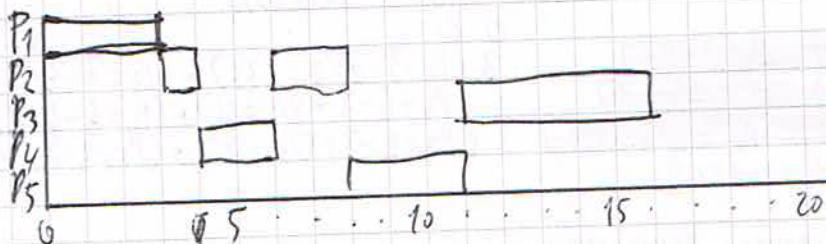
RR
TIME-SLICE = 1 sec.



LCN (MINIMO TEMPO DI CPU) PRIVILEGIA i nuovi processi, raggiungendo quindi esigenze



STG (VERSIONE PRELAZIONATA DELL'SJT)



TEMPO DI ACCESSO EFFETTIVO DELLA MEMORIA:

$$p_{21} \cdot 2 \cdot t_{mem} + (1 - p_{21}) \cdot (t_{mem} + t_{ph} + 2 \cdot t_{mem})$$

LEGENDA

p_{21} : PROBABILITÀ CHE UNA PAGINA ESISTA IN MEMORIA

t_{mem} : TEMPO DI ACCESSO ALLA MEMORIA

t_{ph} : OVERHEAD DI TEMPO NELLA GESTIONE DEL PAGE FAULT.

TEMPO EFFETTIVO DI ACCESSO ALLA MEMORIA (TLB):

$$p_{22} \cdot (t_{TLB} + t_{mem}) + (p_{21} - p_{22}) \cdot (t_{TLB} + 2 \cdot t_{mem}) + \\ (1 - p_{21}) \cdot (t_{TLB} + t_{mem} + t_{ph} + t_{LB} + 2 \cdot t_{mem})$$

LEGENDA

p_{21} : PROBABILITÀ CHE UNA PAGINA ESISTI IN MEMORIA

p_{22} : PROB. CHE UNA ENTRY DI PAGINA ESISTA NELL'TLB

t_{mem} : TEMPO DI ACCESSO IN MEMORIA

t_{LB} : TEMPO DI ACCESSO AL TLB

t_{ph} : OVERHEAD DI TEMPO PER LA GESTIONE DEL PAGE-FAULT

MEMORIA

FIFO : SOSTITUISCE LA PIÙ VECCHIA

LRU : SOSTITUISCE LA MENO RECENZE

CLOCK : RISCA CIRCOLARE CON BIT DI RIFERIMENTO SULLE PAGINE (2 PASSO)

DISPOSITIVI DI I/O

$t_{IO} = \text{TEMPO DI I/O}$

$t_a = \text{TEMPO DI ACCESSO}$

$t_x = \text{TEMPO DI TRASFERIMENTO}$

$$t_{IO} = t_a + t_x$$

NASTRI MAGNETICI

$$t_{IO} = t_a + \frac{s}{v \cdot V}$$

σ_l : DENSITÀ DI MEMORIZZAZIONE

V : VELOCITÀ DEL SUPPORTO DI I/O

DISCO MAGNETICO (HD)

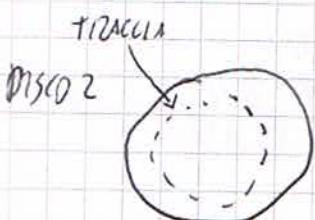
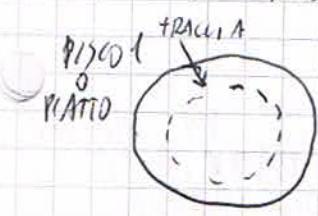
$$t_a = t_s + t_r + t_t$$

t_s : TEMPO DI RICERCA (POSIZIONAMENTO DELLA TESTINA) $5-15 \text{ ms}$

t_r : LATENZA ROTAZIONALE (IN MEDIA 3-4 ms)

t_t : TEMPO DI TRASFERIMENTO (IN MEDIA DECINE DI MEGABYTE AL SEC.)

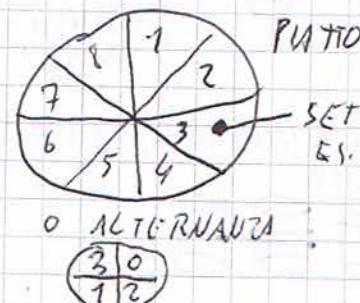
- TUTTE LE TESTINE SONO MONTATE SU UN PICCOLO BRACCIO CHIAMATO ATTUATORE.
- PER AUMENTARE LA CAPACITÀ, POSSONO ESSERE MONTATI PIÙ DISCHI SU UNO STESSO PERNO



L'INSIEME
DELLE STESSE
TRACCE FORMA
UN CILINDRO



INDIRIZZO DI UN RECORD:
(NUM. CILINDRO, NUM. SUPERFICIE, NUM. RECORD)



SETTORI
ES. 10 PIATTI, 9 TRACCE
5 SETTORI:
CILINDRO = 50 SETTORI

- DIMINUIRE I TEMPI DI RICERCA, AUMENTANDO LA VELOCITÀ ROTAZIONALE

RAID

LIVELLO

DESCRIZIONE

0

DISK STRIPING



N DISCHI, INCREMENTO TRA MASSIMO N, I DATI VENGONO SCARICATI SU N DISCHI, SE UNO NON FUNZIONA IL RESTO È INCONCERNENTE.

1

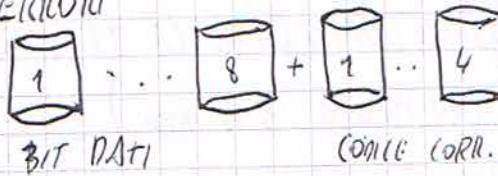
DISK MIRRORING



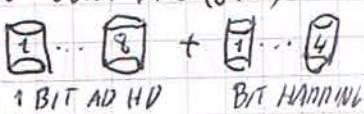
COPIA IDENTICA, AUMENTO DI SPERIMENTO IN LETTURA SE SI ROMPE UN DISCO, TUTTO CONTINUA A FUNZIONARE ALLA PERFEZIONE.

2

COPICE DI CORREZIONE DEGLI ERRORI



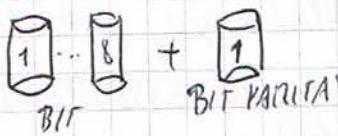
*BIT STRIPING + COPICE DI CORREZIONE
ES. CON HAMMING
PER OGNI BYTE (8 BIT)*



*8 VOL
COST
TROPPI*

3

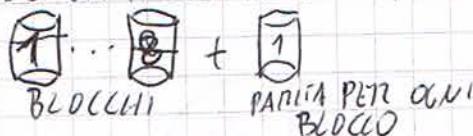
BIT INTERLEAVED PARITY



COME NEL RAID 2 MA UTILIZZA OMOGLIE CONCE DI CORREZIONE UN UNICO BIT DI PARITÀ.

4

BLOCK INTERLEAVED PARITY



COME NEL RAID 3 MA DIVISA A BLOCCHI DI DATI... BYTES DI PARITÀ PER OGNI BLOCCO.

DI PARITÀ

BLOCCHI DI STRIPING SISTEMI TUTTI I DISCHI

5

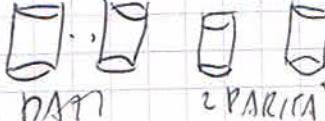
BLOCK INTERLEAVED DISTRIBUTED PARITY



UTILIZZA DUE SCHEMI DI PARITÀ BUILT-IN INDIPENDENTI, PUÒ RIPARARE IL GUASTO DI OGNI DISCO.

6

P+Q REDUNDANCY



~~THROUGHPUT~~

THROUGHPUT DI UN DISCO: NUMERO OPERAZIONI I/O ESEGUITE IN 1 SECONDO.

POLITICHE DI SCHEDULING DEL DISCO:

- FCFS (SCELZIONA L'OPERAZIONE I/O CON TEMPO DI RICHIESTA INFERIORE)
- SSTF (SEL. L'OP. I/O CON IL PIÙ BREVE TEMPO DI RICERCA RISPETTO ALLA POSIZIONE CORRENTE DELLA TESTINA)
- SCAN (SERVIRÀ LE OP. CHE SI TROVANO SU UNA STESSA TRACCIA. QUANDO LA TESTINA VA AVANTI È INPIETTO SERVENDO IN AUMENTO IL RITORNO. UNA VARIANTE LOOK "ASCENSORE" INVERTE LA DIREZIONE SE NON CI SONO PIÙ RICHIESTE)
- CSCAN COME LA SCAN MA SERVIRÀ SOLO IN UN SENSO.

7200 RPM

$$\text{LATENZA ROTAZIONALE} = \frac{60}{2 \cdot 7200} = 4,17 \text{ ms}$$

VELOCITÀ TRASFERIMENTO: $T_t = \frac{b}{\tau \cdot m}$

b : NUM. BLOCCHI DA TRASFERIRE
 N : NUM. BLOCCHI PER TRACCIA
 τ : VELOCITÀ IN RPS

$$\text{TEMPO ACCESSO} \text{ MEDIO} = T_s = \frac{1}{2\tau} + \frac{b}{\tau \cdot N} \quad \begin{array}{l} \text{PER UN TRACCIA} \\ \text{o BLOCCO} \end{array}$$

MTTF + MTRR

~~Tempo di attesa~~:

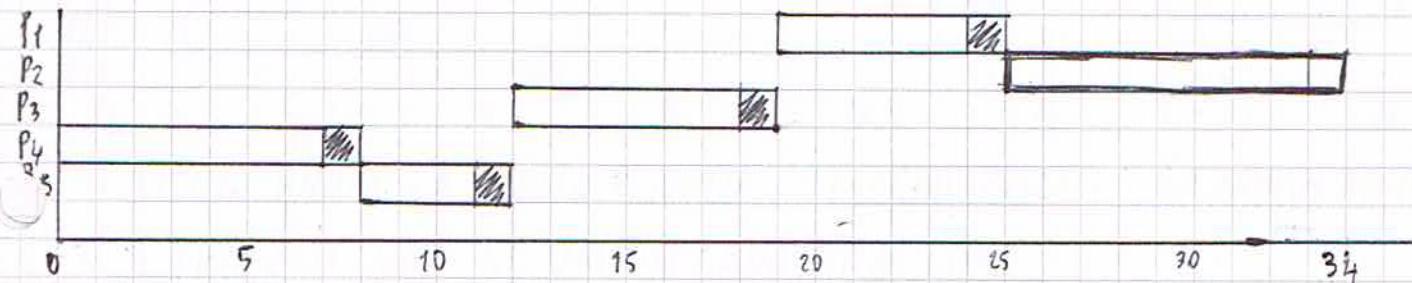
TEMPO TOTALE: TEMPO FINALE (ESEGUITI TUTTI I PROCESSI)

$$\text{THROUGHPUT: } \frac{m}{T_{\text{tot}}} \cdot 1000 = \text{JOB/s}$$

TEMPO DI TORNAROUND

MEDIO:

$$\frac{\text{(SOMMA DEI TEMPI DI OGNI PROCESSO)}}{\text{NUM. PROCESSI}}$$



TEMPO DI ATTESA MEDIO:

~~Tempo di attesa~~:

TEMPO DI TORNAROUND MEDIO:

$$(7 + 11 + 11 + 24 + 34) / 5 = 18,8 \text{ msr}$$

TEMPO DI TORNAROUND NORMALIZZATO MEDIO:

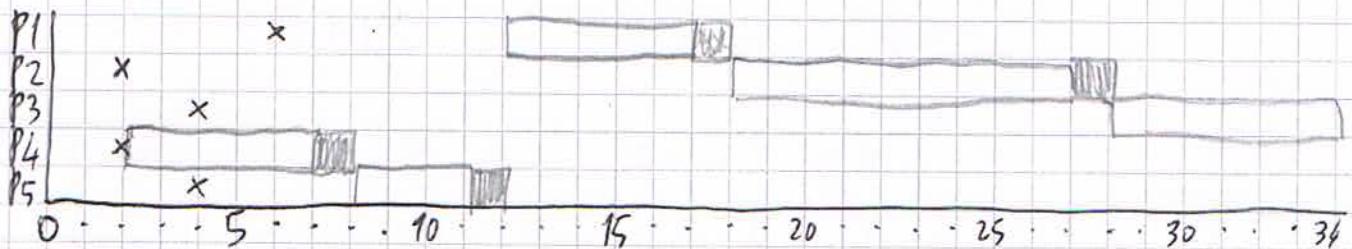
$$\left[\frac{(5+19)}{5} \right] + \left[\frac{(9+25)}{5} \right] + \left[\frac{(6+12)}{5} \right] + \left[\frac{(0+7)}{5} \right] + \left[\frac{(8+3)}{5} \right] = \\ -4,8 + 6,8 + 3,6 + 1,4 + 2,2 = 18,8 \text{ msr}$$

$$\text{THROUGHPUT: } \frac{m}{T_{\text{tot}}} \cdot 1000 = \frac{5}{34} \cdot 1000 = 147,05 \text{ JOB/s}$$

$$\text{TEMPO TOTALE: } 34 \text{ SEC}$$

	T. ARRIVO	CPU-BURST	PRIORITA'
P1	6 msr	5	2
P2	2 "	9	1
P3	4 "	6	1
P4	2 "	5	5
P5	4 "	3	4

PRIORITA' CON PRELAZIONE - CAMBIO CONTESTO 1 msr - PRIORITA' MAX 5

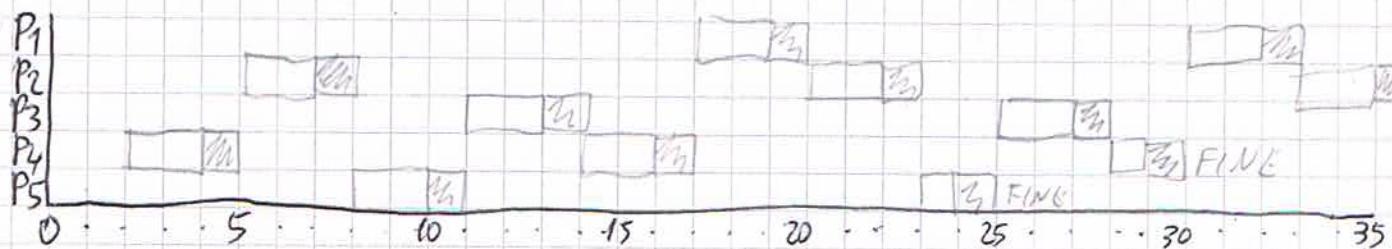


- 1) P2 E P3 HANNO LA STESSA PRIORITA'... ESEGUE COLUI CHE ARRIVA PRIMA.
- 2) I PROCESSI NON SI DISTURBANO TRA ZORO

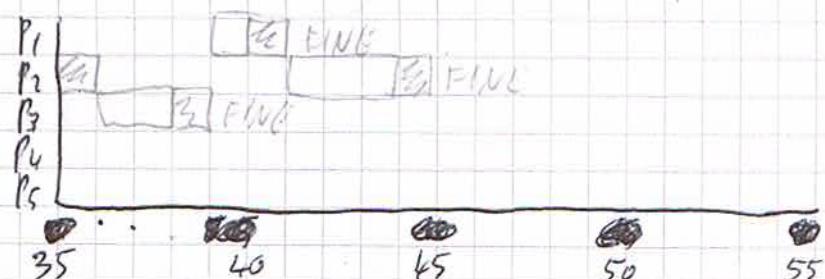
$$\text{TEMPO TOTALE} = 34 \text{ ms}$$

$$\text{TEMPO DI ATTESA MEDIO} = \frac{[(12-6)+(18-2)+(28-4)+(2-2)+(8-4)]}{5} = \\ (6+16+24+0+4)/5 = 10 \text{ ms}$$

RR - CAMBIO CONTESTO 1 msr - QUANTUM Q = 2 ms



5	9	6	5	3
P1	P2	P3	P4	P5
2	2	2	2	2
7	3	2	7	1
7	2	2	7	1
11	11	11	11	11



TEMPO TOTALE = 45 ms
TEMPO MEDIO D'ATTESA:

SOSTITUZIONE PAGING

PAG. 471

5, 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5

FIFO

5	2	2	2	3	3	3	3	1	1		
4	4	1	1	1	5	5	5	5	5	5	
3	3	3	4	4	4	4	4	2	2	2	
2*	1*	4*	3*	5*	4	3	2*	1*	5		

7 FP

LRU

5	2	2	2	3	3	3	3	3	3	5	
4	4	1	1	1	5	5	5	2	2	2	
3	3	3	4	4	4	4	4	1	1	1	
2*	1*	4*	3*	5*	4	3	2*	1*	5*		

8 FP

CLOCK

5*	2*	2*	2*	3*	3*	3*	3*	1	1		
4*	4	1*	1*	1	5*	5*	5*	5	5	5	
3*	3	3	4*	4	4	4*	4*	2*	2*	2*	
2*	1*	4*	3*	5*	4	3	2*	1*	5		

7 FP

3/3/2008

FRAME N 512

PAGINE 512

1 NO GM PAGINA POSSO METTERE 512/4 = 128

A [2] [N]	⊕ PAGINE
B [n]	4 "
C [3N]	12 "

NUM PAGE

0	A [0] [0 .. 127]
1	A [0] [128 .. 255]
2	A [0] [256 .. 383]
3	A [0] [384 .. 511]
4	A [1] [0 .. 127]
5	A [1] [128 .. 255]
6	A [1] [256 .. 383]
7	A [1] [384 .. 511]
8	B [0 .. 127]
9	B [128 .. 255]
10	B [256 .. 383]
11	B [384 .. 511]
12	C [0 .. 127]
13	C [128 .. 255]
14	C []
15	C []
16	C []
17	C []
18	C []
19	C []
20	C []
21	C []
22	C []
23	C []
24	C []

A [0] [0 .. 127]
A [0] [128 .. 255]
A [0] [256 .. 383]
A [0] [384 .. 511]
A [1] [0 .. 127]
A [1] [128 .. 255]
A [1] [256 .. 383]
A [1] [384 .. 511]
B [0 .. 127]
B [128 .. 255]
B [256 .. 383]
B [384 .. 511]
C [0 .. 127]
C [128 .. 255]

$$* ab = c$$

$0 \leq j < 66$	$66 \leq j < 128$	$128 \leq j < 192$	$192 \leq j < 256$
PER $i=0$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 0 & 8 & 12 \\ \hline 1 & 10 & 14 \\ \hline 2 & 12 & 16 \\ \hline 3 & 14 & 18 \\ \hline 4 & 16 & 20 \\ \hline 5 & 18 & 22 \\ \hline 6 & 20 & 24 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 2 & 8 & 14 \\ \hline 3 & 10 & 16 \\ \hline 4 & 12 & 18 \\ \hline 5 & 14 & 20 \\ \hline 6 & 16 & 22 \\ \hline 7 & 18 & 24 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 3 & 8 & 15 \\ \hline 4 & 10 & 17 \\ \hline 5 & 12 & 19 \\ \hline 6 & 14 & 21 \\ \hline 7 & 16 & 23 \\ \hline \end{array}$
$i=1$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 4 & 8 & 12 \\ \hline 5 & 10 & 14 \\ \hline 6 & 12 & 16 \\ \hline 7 & 14 & 18 \\ \hline 8 & 16 & 20 \\ \hline 9 & 18 & 22 \\ \hline 10 & 20 & 24 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 6 & 8 & 14 \\ \hline 7 & 10 & 16 \\ \hline 8 & 12 & 18 \\ \hline 9 & 14 & 20 \\ \hline 10 & 16 & 22 \\ \hline 11 & 18 & 24 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline A & B & C \\ \hline 7 & 8 & 15 \\ \hline 8 & 10 & 17 \\ \hline 9 & 12 & 19 \\ \hline 10 & 14 & 21 \\ \hline 11 & 16 & 23 \\ \hline \end{array}$

0 8 12 1 8 13 2 8 14 3 8 15 4 8 12 5 8-13 6-8-16-7-8-15

PAGINE 512 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

25/9/08

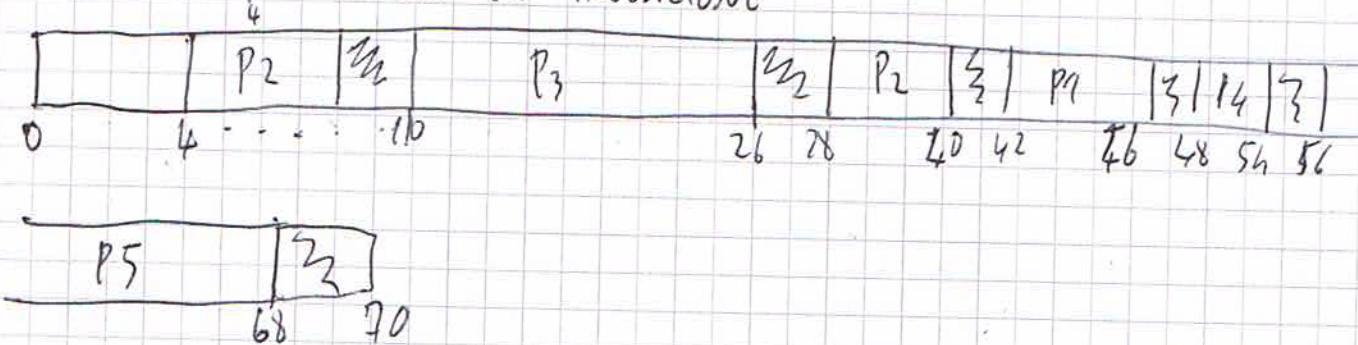
SCHEDULING.

PROCESSO	TEMPO DI ARRIVO	CPU-BURST	PRIORITA'
P ₁	8	6	3
P ₂	4	11	2
P ₃	8	16	1
P ₄	12	6	4
P ₅	18	12	5

$$CS = 2 \text{ ms}$$

PRIORITA' CON PRELAZIONE

PRIORITA MAX 1



$$T. ATTESA MENO \left[(28-8) + (-4+6) \right]$$

$$(42-8) + (12-8) + (10-8) + (6-8) + (56-18) = \underline{34 + 20 + 8 + 36 + 38} - 26 \quad 5$$

T. TORNAROUND

T. TORNAROUND MEDIO

$$\text{THROUGHPUT} = \frac{5}{68} =$$

17/9/2010

PROCESSO A

COPICE A	DATA A	STACK A
24000	26999	41000 49999
REC. BASE	REC. LIMITE	REC. LIMITE
21000	26000	62050 66049
REC. LIMITE	REC. LIMITE	REC. LIMITE
8000	9000	4000

PARTIZ. LIBERE

9000	11000	15000
70000	78999	300000 374999

PROCESSO B

BEST FIT

FREE	FREE	FREE	FREE
70000	77999	90000	300000 308999

COPICE B

DATA B

STACK B

REG. BASE	REG. LIMITE	REG. LIMITE
100000	90000	300000
REC. LIMITE	REC. LIMITE	REC. LIMITE
8000	9000	2000

PARTIZ. LIBERE

REG. BASE	REG. LIMITE
70000	100000
REG. BASE	REG. LIMITE
90000	20000

REG. BASE	REG. LIMITE
300000	110000

FIRST FIT E NEXT FIT IN QUESTO CASO RISULTANO uguali alla BEST FIT.

WORST

	STACK B FREE	PATI B FREE	COPICE B FREE
5000			
70000	73799	90000	98777
COPICE BASE	PATI BASE	STACK BASE	
300000	90000	70000	
COPICE LIM.	7171 LIM.	STACK LIM/175	
8000	9000	4000	

STAZIO LIBERO

REC. BASE	76000	99.000	307999
REC. LIMITE	5000	2.000	7000

$$\frac{\text{DISCHI} \quad \text{non} \quad \text{OK} \quad \text{GIAJIS}}{\text{DISCHI} \quad \text{R01T1}} =$$

$$2 \cdot \frac{1000}{4} = 2000$$

$$4 \cdot \frac{1000}{2} =$$

MTTF

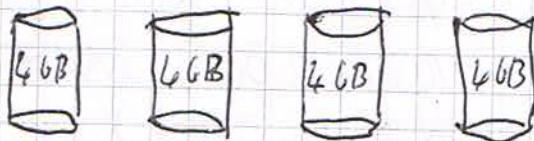
$$\text{RAID 0} = \frac{\text{MTTF DISCO}}{N}$$

$$\text{RAID 1} = \frac{(\text{MTTF DISCO})^2}{2 \cdot N \cdot \text{MTTR}}$$

$$\text{RAID 2 AS} = \frac{(\text{MTTF DISCO})^2}{N \cdot (N-1) \cdot \text{MTTR}}$$

RAID

LVR-0 16 GB STRIP 512 KB



LVR-1 8GB 8GB 8GB 8GB 1N 4 DISC(1)
 |
 |

Diagram illustrating RAID 1 (Mirroring). Four individual drives, each labeled 8GB, are shown being mirrored into two mirrored volumes. Each mirrored volume has a capacity of 16 GB, and the total capacity is 32 GB. The label "1N 4 DISC(1)" indicates that there are four disk groups, each containing one drive from each pair.

PROVA 24/9/07

$$T_{TLB} = 150 \text{ ms}$$

$$T_{MEM} = 400 \text{ ms} \quad \text{DEVE ESSERE SEMPRE MAGGIORE}$$

$$T_{PFA} = 8 \text{ ms} \Rightarrow 8.000.000 \text{ ms}$$

$$\text{PAGE FAULT RATE} \rightarrow (1 - PR_1) = 0,02 \text{ (2\%)} \\ T_{LB} \text{ HIT RATE} (PR_2) = 0,7 \text{ (7\%)}$$

$$\underline{\text{EAT DATI}} = 0,7(150 + 400) + (0,98 - 0,7)(150 + 800) - \\ (0,02)(150 + 400 + 150 + 800 + 8.000.000) = 16086$$

(2)

$$T_{LB} \text{ MISS RATE} (PR_1 - PR_2 = 0,1) 10\% \quad (1 - PR_1) = 0,04 \\ \text{6 VERSO 2\% DEGLI INNANZI LOCICI (AVIA) UN P\%} \\ \text{DIRTY BIT } 0 = 50\% \text{ (DEC 4\%)}$$

$$T_{SWAP} = 1 \text{ ms}$$

$$\text{EAT} = 0,86(150 + 400) + 0,1 \cdot (150 + 800) + (0,04) \cdot \\ (150 + 400 + 150 + 800 + 8.000.000) + 0,04 \cdot 0,5 \\ (150 + 400 + 150 + 800 + 8.000.000 + 1.000.000) = 340628$$

①

++

$$T_{TLB} = 15 \text{ ms}$$

$$T_{MEM} = 70 \text{ ms}$$

$$T_{PFA} = 50 \text{ ms}$$

$$\text{PAGE FAULT RATE} (1 - PR_1) = 5\% \quad (0,05)$$

$$T_{LB} \text{ HIT RATE} (PR_2) = 60\% \quad (0,6)$$

$$\underline{\text{EAT DATI}} = 0,6(15 + 70) + (0,95 - 0,6) \cdot (15 + 70) \\ + 0,05(15 + 70 + 15 + 70 + 5.000.000) =$$

2500117,25 ms

28/6/07

~~1000-2000~~

$$1 \text{ BLOCO} = 512 \text{ B}$$

$$N = 256$$

$$A = [0 \dots 255] = 4 \text{ PAGE}$$

$$C = [3][N] = 6 \text{ PAGE}$$

$$1 \text{ INTERO} = 4 \text{ B}$$

$$512 \text{ B} / 4 \text{ B} \neq 128.$$

$$N = 256 = 2 \text{ PAGE}$$

$$A[0 \dots 255] = 4 \text{ PAGE}$$

- 0 A[0 .. 127]
- 1 A[128 .. 255]
- 2 A[256 .. 383]
- 3 A[384 .. 511]

$$4 C[0][0 \dots 127]^5 [128 \dots 255] \quad \text{circled}$$

$$5 C[1][0 \dots 127]^7 [128 \dots 255]$$

$$6 C[2][0 \dots 127]^9 [128 \dots 255]$$

NOTAZIONE POLARE

$$+QC = C$$

~~0 1 2 3 4 5 6 7 8 9 10 11 12 13~~

$$0 < J < 31 \quad 32 < J < 63 \quad 64 < J < 95 \quad 96 < J < 127$$

I ₀	0 5 4	1 5 4	2 5 4	3 5 4
I ₁	0 7 6	1 7 6	2 7 6	3 7 6
I ₂	0 9 8	1 9 8	2 9 8	3 9 8

TAB. RISORSE

26/6/07

	R ₁	R ₂	R ₃	R ₄
P ₁	2	0	1	0
P ₂	0	3	8	2
P ₃	1	1	8	0
P ₄	0	0	2	2

TABELLA NEED

	R ₁	R ₂	R ₃	R ₄
P ₁	0	2	0	0
P ₂	2	0	8	0
P ₃	0	0	0	0
P ₄	0	0	1	0

RIS. DISPONIBILI

R ₁	R ₂	R ₃	R ₄
O	0	0	0

A) Sistema non è in uno stato sicuro.

B) Seppellendo P₄ è sicuro?

RIS. DISP.

R ₁	R ₂	R ₃	R ₄
O	0	2	2

B) Continua ad essere non sicuro.

FILE = 200 RECORD

MENORIZZATI 15 PER BLOCCO

QUANTI ACCESI MEMORIA E' NCESSARIO PER CANCELLARE IL 121-ESIMO RECORD?

- 1) ULTIMO RECORD LOTTO: 34 (CONTIGUA, CONSECUTIVA E CON TAB. IN P)
- 2) FILE SYSTEM VM/X 4KB BLOCCO, QUANTI BLOCCON PER SCRIVERE 700 BYTES,
ALL'OFFSET 120⁹? QUALE DIMENSIONE AVRA' IL FILE AL TERMINE DELLA SCR.

$$200/15 = 13,3 \text{ ovvero } 14 \text{ BLOCCON}$$

0..	14
15..	29
30	44
45	59
60	74

$$\text{POICHE } (121-34)/15 = 5,8$$

IL 121 STA AL 6 BLOCCO
DOPPIO IL 34 ESISTE RECON
 $34/15 = 2,26 = 3 \text{ BLOCCO}$

16 ACCESSI PLSIO

1 MEN.

QUINDI STO AC TORIZZO
BLOCCO 6 DEVO ARRIVARE
AC 9 (3+6).

3/3/2008

FILE SYSTEM 16 BLOCCHI ED USA LA TECNICA
DELLA BITMAP

1000 1100 0000 1010

- a) Scrivere nel file A (4 BLOCCHI)
- b) Scrivere nel file B (3 BLOCCHI)
- c) Aggiunre di due blocchi ed a cancellazione del B

1000 1100 0000 1010
||
OCCUPI

LO STATO DEL BLOCCO 0 E' RAPPRESENTATO DAL B/L
PIU' A SINISTRA

FS ALLOCHE SEMPRE 1 BLOCCO DI INIZIO MINORE

- a) SCRIVERE NEL FILE A (4 BLOCCHI)
CONTIGUA

1000 111111001010
AAAAAA

- b) SCRIVERE DI B (3 BLOCCHI)

1111111111001010
BBB AAAA

- c) AGGIUNTA DI 2 BLOCCHI AP A, CANCELLAZIONE DI B.

1111111111010
BBB AAAA

10001111111010
AAA AAA

(CONCATENATA)

1000 1100 0000 1010
SCRITTO A (4 BLOCCHI)

1111111000001010
AAA A

b) SCRITTO B (3)

1111111001010
AAA A BBB

c) AGGIUNTA DI 2 BLOCCHI ADA, CANCELLAZIONE DI B

1111111111111010
AAA A BBBB AA

111111100011010
AAA A AA

INDICIZZATA

1000 1100 | 0000 1010

SCRITTO DI A (4 BLOCCHI) \rightarrow SCRITTO B (3 BLOCCHI)

11111111111010
AA AA BBB

AGGIUNTA E CANCELLAZIONE

11111110000111
AA AA AA AA



FILE SYSTEM

DISCO COMPOSTO 300 CILINDRI
 OGNI CILINDRO CONTIENE 296 BLOCCHI
 OGNI BLOCCO È DI 4 KB

RICHIESTA DI BLOCCHI LOGICI = 5-10-15-530-2065-9320
 4660-6320

CILINDRO INIZIALE: 280

OGNI BLOCCO 4096 BYTE

INDIRIZZO DA 32 BIT (4 BYTE)

$\frac{4096}{4} = 1024$ (OGNI BLOCCO INDICE PUÒ PUNZARE A 1024 B.DAT)

B. INDICE	1012 DATI	BLOCCO 13 IND. 1	14 & 1038 DATI	BLOCCO 1039 IND. 2	BLOCCO 1060 IND. 2-1	1064 & 1065 DATI
2066 IND. 2-2	DA 2067-3091 DATI	3092 IND. 2-3	3093-4117 DATI	4112 IND. 2-4	4113 & 5163 DATI	5164 IND. 2-5
6730 IND. 2-6	6771 & 7195 DATI	7196 IND. 2-7	7197-8220 DATI	8221 IND. 8	8222-9246 DATI	9247 IND. 7

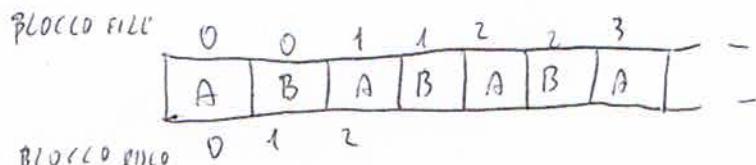
TRASFORMAZIONE INDIR. DA LOGICO A FISICO E DA FIS. A CILINDRO

BLOCCO	5 = 1	BASE + 5 =	6 / 296 = 0 CILINDRO
"	10 = 1	" + 10 =	11 / 296 = 0 "
"	15 = 2	" + 15 =	17 / 296 = 0 "
"	530 = 2	" + 530 =	532 / 296 = 1 "
"	2065 = 4	" + 2065 =	2069 / 296 = 7 "
"	9340 = 12	" + 9340 =	9352 / 296 = 31 "
"	4540 = 7	" + 4540 =	4567 / 296 = 15 "
"	6320 = 9	" + 6320 =	6329 / 296 = 21 "
"	1070 = 2	" + 1070 =	1072 / 296 = 3 "
"	35 = 2	" + 35 =	37 / 296 = 0 "

FILE A & B

(5, A) (10, A) (15, A) (530, B) (2065, A) (8340, B)
 (4540, B) (6320, B)

FILE A & B ALLOCATI OGNI 2 BLOCCHI A PARTIRE DA 0.



$$A = \text{PAIR} = N_{\text{TRACCIA}} \cdot 2$$

$$B = b_{\text{PAIR}} = (N_{\text{TRACCIA}} \cdot 2) - 1$$

DISCO 400 CILINDRI (0, 399)

1 CILINDRO = 260 BLOCCHI DA 4 KB

CINDRIZZI = 32 BIT

TRANSF. 10 MBITS

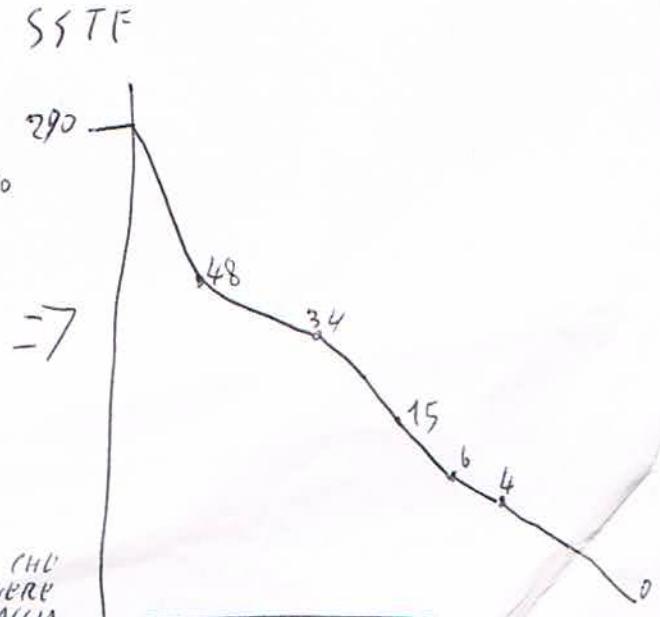
$$\text{LATENZA MASSIMA} = \frac{1}{2} = \frac{1}{120} = 8,3 \text{ ms}$$

ULTIMO SPOST. TRACCIA 290

SSTF? C-SCAN?

$$\begin{aligned}
 5 \cdot 2 &= 10 \quad /260 = 0 \quad (\text{CILINDRO FISICO}) \\
 10 \cdot 2 &= 20 \quad /" = 0 \quad " \\
 (15 \cdot 2) &= 30 \quad /" = 0 \quad " \\
 (530 \cdot 2) - 1 &= 1059 \quad /" = 4 \quad " \\
 (2065 \cdot 2) &= 4130 \quad /" = 15 \quad " \\
 (834 \cdot 2) - 1 &= 1667 \quad /" = 6 \quad " \\
 (4540 \cdot 2) - 1 &= 9079 \quad /" = 34 \quad " \\
 (6320 \cdot 2) - 1 &= 12639 \quad /" = 48 \quad "
 \end{aligned}$$

✓ BLOCCHI CHI
VA A LEGGERE
SULLA TRACCIA



$$T_{290-48} = 242 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{1}{260}\right) = 56,73 \text{ ms}$$

$$T_{48-34} = 12 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{1}{260}\right) = 10,73 \text{ ms}$$

$$T_{34-15} = 19 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{1}{260}\right) = 12,13 \text{ ms}$$

$$T_{15-6} = 9 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{1}{260}\right) = 10,13 \text{ ms}$$

$$T_{6-4} = 2 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{1}{260}\right) = 8,73 \text{ ms}$$

$$T_{4-4} = 4 \cdot 0,2 + 8,3 + \left(8,3 \cdot \frac{3}{260}\right) = 9,19 \text{ ms}$$

PERCHE' 3 BLOCCHI SUL CILINDRO 0

23/7/2010

P₁ P₂ P₃

FILE A e B

(5, A) (10, A) (15, A) (530, B) (2065, A) (8340, B)
(4540, B) (6320, B)

FILE A e B HANNO BLOCCHI FISICI ALLOCATI OGNI 2
A PARTIRE DA 0 PER IL FILE A E DA 1 PER
IL FILE B

DISCO = 400 CILINDRI (0..399)

1 CILINDRO = 260 BLOCCHI DI 4 KB

INDIRIZZI A 32 BIT

TRASFERIMENTO 10 MB/s

CHE LA LATENZA SIA MASSIMA

ULTIMO SPOSTAMENTO DAL CILINDRO 230 A 290

- TEMPO NECESSARIO PER ACCEDERE ALLE TRACCE

- STOP

- SCAN se ARRIVA A 10 ms LA RICHIESTA DI LETTURA DEL BLOCCO

EAT ✓ HIT RATE

new page

$$PR_2(T_{LB} + T_{Mem}) + (PR_1 - PR_2)(T_{TLB} + 2T_{Mem}) + (1 - PR_1) \cdot \\ \cdot (3T_{Mem} + 2T_{TLB} + T_{PF})$$

DRRU(2)

$$(1 - PR_1)(\text{perc. DIRTY BIT})(2T_{LB} + 3T_{Mem} + T_{PF}) + \\ (1 - PR_1)(\text{perc. RESUME PERTURB})(2T_{LB} + 3T_{Mem} + T_{PF} + t_{swap})$$

↑
PAGE FAULT
BIT RATE

25/9/2008

$$TLB = 150 \text{ ns}$$

$$REN = 400 \text{ ns}$$

$$\text{PAGE FAULT} = 5 \text{ ns} \quad (5 \cdot 000 \cdot 000 \text{ ns})$$

$$\text{PF RATE} = 5\% = (0,05) = (1 - PR_1) = 0,95$$

$$TLB \text{ HIT RATE} = 10\% (0,6)$$

A)

$$\text{EAT: } 0,6(150 + 400) + (0,95 - 0,6)(150 + 800) + (0,05)(100 \cdot 100 \cdot 100) \\ (1200 + 300 + 5 \cdot 000 \cdot 000) = 2507375$$

$$B) (PR_1 - PR_2) = 20\% = 0,2$$

$$(1 - PR_1) = 3\% = (0,03) \quad (1 - 0,03) = 0,97 \quad PR_2 = 0,77 \text{ ns}$$

$$\text{PF RATE } 40\% = 0,04 \quad (\text{DIRTY BIT } 10) \quad (80\% \text{ BIT } 11)$$

$$t_{swap} = 2 \text{ ns} = (2 \cdot 000 \cdot 000) \text{ ns}$$

$$0,77(150 \text{ ns} + 400) + (0,2)(150 + 800)(0,03)(0,04) \cdot 100 \cdot 100 \cdot 100$$

$$(300 + 1200 + 5 \cdot 000 \cdot 000) + (0,03)(0,6)(300 + 1200 + 5 \cdot 000 \cdot 000 + 2000000)$$

$$= 186.658,5$$

DISCO 3/7/09

$$T_s + \frac{1}{22} + \frac{b}{7N} \geq 2 + \frac{1}{44}$$

7000 CILINDRI

20 TRACCE = 1 CILINDRO

160 SETTORI PER TRACCIA

DIM. SETTORE 512B

T. SEEK 2 ms (PER OGNI CILINDRO ATTRAVERSATO)

T. ATTRAVERS. TRA 1 CILINDRO 0,5 ms

F. RIVOLUZIONE = 7200 RPM $\rightarrow \frac{60}{22} = \frac{60}{\frac{14400}{100}} = 4,17 \text{ ms}$

TESTINA SU CILINDRO 100

ATTESA MEDIA = $\frac{1}{2}$ ROTAZIONE

86 1470 913 1774 948 1509 1022 1750 130

|100



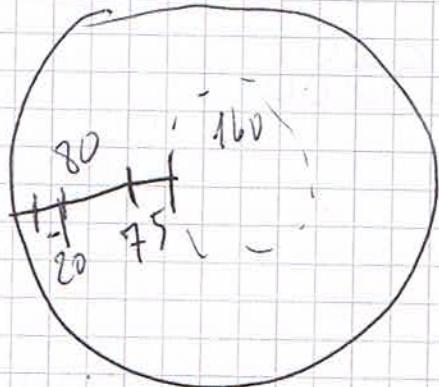
$$(430 - 100) \left| \left(\frac{160}{2 \cdot 7200} \right) \right| \left\{ b = \frac{1}{2} \right.$$

$$2 \cdot 30 + 4,1 + \left(\frac{4}{160} \cdot 8,3 \right) = 64,3075$$

$$2(913 - 130) + \frac{1}{2(7200)} + \left(\frac{4}{160} \cdot 8,3 \right) = 1566 + 4,1 + 0,2 = 1570,3$$

7200

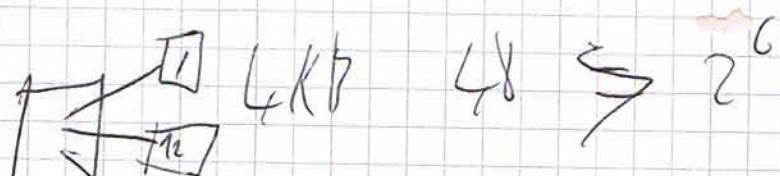
5600



$$S = 4$$

$$BGB = 8$$

$$(4 \cdot 55) + \left(\frac{10}{160}\right) + \left(\frac{8}{160} \cdot 8,3\right) = 220 + 6,1 + 0,04 = 226,15$$



$$2^{10} \cdot 2^{12} = 2^{22}$$

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} = 2^{20} \cdot 2^{12} = 2^{32}$$

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} = 2^{10} \cdot 2^{12} = 2^{42}$$

$$2^{96} = 96/3 = 32$$

$$1024 \text{ Byte} = 2^{10}$$

$$1024 \text{ KB} = 2^{10}$$

FAT

BLOCKCH = 16 BYTES

████████████

$$512.000 / 16 = \cancel{32768} \quad 32768 \text{ BLOCKCH (16K)}$$

bytes 125384.

$$125384 / 16 = 122, = 7,6 = \frac{7}{7} \text{ BLOCKS}$$

1 KB

7 blocks

BANCHIERE

A L F O SATE
A B C D

P ₀	6	0	$Z+4$	2
P ₁	10	0	0	2
P ₂	6	$2-x$	0	0
P ₃	0	0	3	2
P ₄	4	1	$2-y$	4

R. MASSIMO

A	B	C	D
8	4	5	6
12	7	6	8
8	2	0	8
0	3	4	2
19	9	6	9

R. DISOMBILU

A	B	C	D
6	4	10	4

NEGO

P ₀	2	4	$1-z$	4
P ₁	2	7	6	6
P ₂	2	X	8	8
P ₃	0	3	7	9
P ₄	0	0	$4+y$	5

$$\begin{array}{cccc} \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} \\ b & 12 & 4 & Z+14 \end{array} \quad 6$$

$$\left\{ \begin{array}{l} Z+4 \geq 0 \\ 1-z \geq 0 \\ 10 \geq 1-z \\ Z \leq y \end{array} \right. \quad \left. \begin{array}{l} Z \geq -4 \\ Z \leq 1 \\ Z \geq y \end{array} \right.$$

$$-4 \leq Z \leq 1$$

NON CONSIDERO \downarrow

$$\left\{ \begin{array}{l} Z+14 \geq 0 \\ Z+14 \leq 6 \end{array} \right. \quad \left\{ \begin{array}{l} Z \geq -14 \\ Z \leq -8 \end{array} \right.$$

$$P_1 22 \quad 4 \quad Z+14 \quad 8$$

$$\left\{ \begin{array}{l} X \geq 0 \\ 2-x \geq 0 \\ 4 \geq x \end{array} \right. \quad \left\{ \begin{array}{l} X \geq 0 \\ -x \geq -2 \\ x \leq 4 \end{array} \right. \quad \left\{ \begin{array}{l} X \geq 0 \\ X \leq 2 \\ X \leq 4 \end{array} \right.$$

$$0 \leq x \leq 2$$

$$P_3 28 \quad 6-x \quad Z+17 \quad 10$$

$$\left\{ \begin{array}{l} 6-x \geq 3 \\ Z+17 \geq 10 \end{array} \right. \quad \left\{ \begin{array}{l} -x \geq -3 \\ Z \geq -13 \end{array} \right. \quad \left\{ \begin{array}{l} X \leq 3 \\ Z \geq -13 \end{array} \right.$$

$$P_4 32 \quad 7-x \quad Z+11 \quad 14$$

$$\left\{ \begin{array}{l} 2-y \geq 8 \\ 4+y \geq 8 \\ Z+11 \geq 4+y \end{array} \right. \quad \left\{ \begin{array}{l} -y \geq -2 \\ y \geq -4 \\ y = 13 + z \end{array} \right. \quad \left\{ \begin{array}{l} Y \leq 2 \\ Y \geq -4 \\ Y = 13 + z \end{array} \right.$$

$$-4 \leq Y \leq 2$$

LA SEQ. P_0, P_1, P_2, P_3, P_4 E' SI CURA.

Osservazione n° 4

Ad un disco rigido composto da 200 cilindri (da 0 a 199) e 20 settori per traccia, vengono richiesti i seguenti spostamenti (il numero indica il numero di cilindro richiesto) per trasferire un numero di settori riportato in parentesi: 70 ($s=2$), 40 ($s=4$), 155 ($s=5$), 90 ($s=7$), 130 ($s=3$), 10 ($s=2$), 120 ($s=3$), 17 ($s=4$). Supponendo che l'inversione di movimento richieda 500 μ s, che lo spostamento da un cilindro al cilindro vicino richieda 10 μ s, che il tempo di latenza massimo sia 2ms, si calcoli il tempo necessario per l'accesso alle tracce secondo gli algoritmi SSTF, SCAN, ~~LOOK~~. L'ultima posizione della testina è il cilindro 75 e si stava spostando verso lo 0.

TEMPO DI ATTENZIA \rightarrow

$$T_E = \frac{1}{2r}$$

CASO MEDIO \rightarrow $\frac{1}{r}$ CASO PEGGIORE ?

$r \rightarrow$ rps (VELOCITÀ DI ROTAZIONE)

TEMPO DI RICERCA (SEEK TIME) \rightarrow

$$T_S = m \cdot m$$

$m \rightarrow$ TEMPO DI SPOSTAMENTO DI UNA TRACCIA (CILINDRO)

$m \rightarrow$ NUMERO DI TRACCE (CILINDRI) CHE ATTRAVERSA

TEMPO DI TRASFERIMENTO \rightarrow

$$T = \frac{b}{r \cdot N}$$

$b \rightarrow$ NUBERO DI INFORMAZIONI (SETTORI o byte) DA LEGGERE

$N \rightarrow$ NUBERO TOTALE DI INFORMAZIONI PRESENTI SU UNA TRACCIA (CILINDRO)

TEMPO DI ACCESSO MEDIO

$$T_{AM} = T_S + T_E + T = m \cdot m + \frac{1}{r} + \left(\frac{1}{r} \cdot \frac{b}{N} \right)$$

$$m \rightarrow 10 \text{ ns} \rightarrow 0,01 \text{ ms}$$

$$b \rightarrow s \cancel{\text{ms}}$$

$$m \rightarrow$$

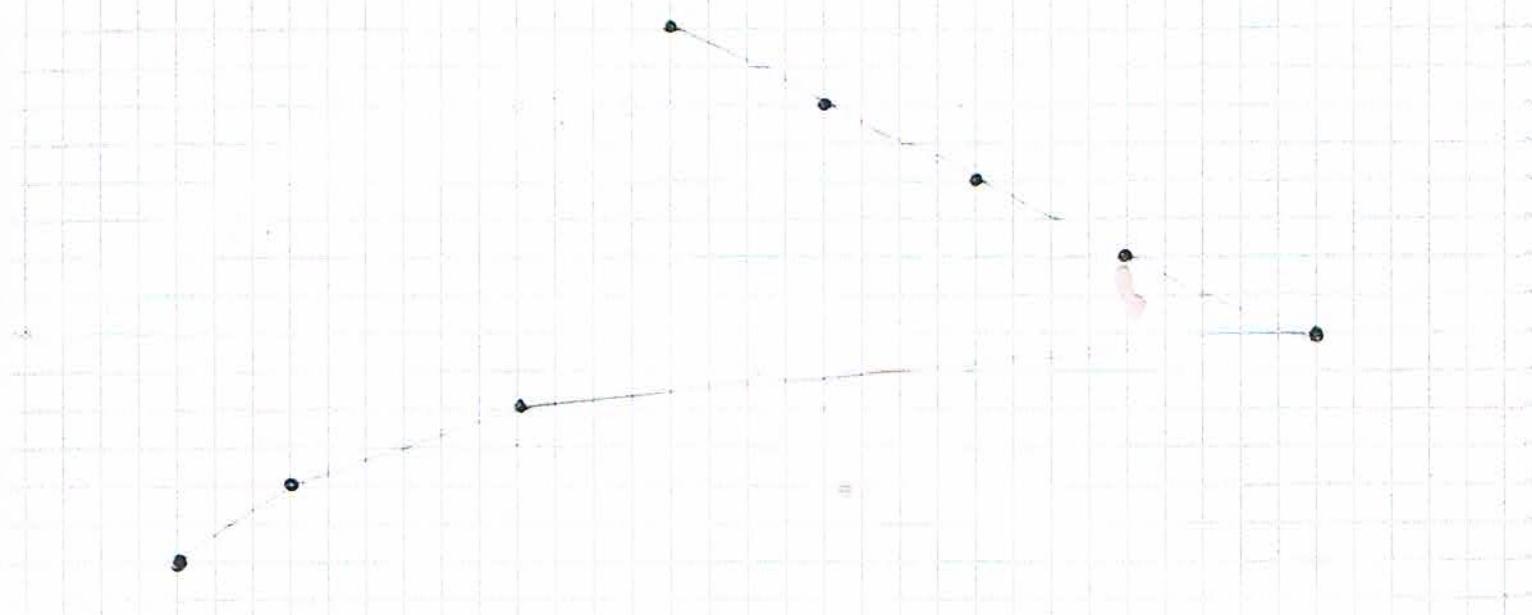
$$N \rightarrow 20$$

$$T_E \rightarrow 2 \text{ ms}$$

$$\text{INVERSIONE} \rightarrow 500 \text{ ps} \rightarrow 5 \text{ ms}$$

APPLICO LA SSTF

0	10	17	40	70	75	90	120	130	155	199
1	1	1	1	1	1	1	1	1	1	1



$$T_{75 \rightarrow 70} = 0,01 \cdot 5 + 2 + \left(2 \cdot \frac{3}{20} \right) = 0,05 + 2 + 0,2 = 2,25$$

$$T_{70 \rightarrow 90} = 0,01 \cdot 20 + 2 + \left(2 \cdot \frac{7}{20} \right) = 0,2 + 2 + 0,7 = 2,9$$

$$T_{90 \rightarrow 120} = 0,01 \cdot 30 + 2 + \left(2 \cdot \frac{3}{20} \right) = 0,3 + 2 + 0,3 = 2,6$$

$$T_{120 \rightarrow 130} = 0,01 \cdot 10 + 2 + \left(2 \cdot \frac{3}{20} \right) = 0,1 + 2 + 0,3 = 2,4$$

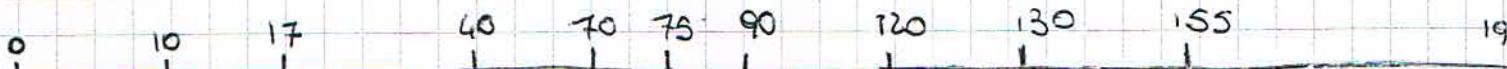
$$T_{130 \rightarrow 155} = 0,01 \cdot 25 + 2 + \left(2 \cdot \frac{5}{20} \right) = 0,25 + 2 + 0,5 = 2,57$$

$$T_{155 \rightarrow 40} = 0,01 \cdot 115 + 2 + \left(2 \cdot \frac{4}{20} \right) + 5 = 1,15 + 7 + 0,4 = 8,55$$

$$T_{40 \rightarrow 17} = 0,01 \cdot 23 + 2 + \left(2 \cdot \frac{4}{20} \right) = 0,23 + 2 + 0,4 = 2,63$$

$$T_{17 \rightarrow 10} = 0,01 \cdot 7 + 2 + \left(2 \cdot \frac{2}{20} \right) = 0,07 + 2 + 0,2 = 2,27$$

APPLICO LA SCAN



$$T_{75 \rightarrow 70} = 0,01 \cdot 5 + 2 + \left(2 \cdot \frac{2}{20} \right) = 0,05 + 2 + 0,2 = 2,25$$

$$T_{70 \rightarrow 40} = 0,01 \cdot 30 + 2 + \left(2 \cdot \frac{4}{20} \right) = 0,3 + 2 + 0,4 = 2,7$$

$$T_{40 \rightarrow 17} = 0,01 \cdot 23 + 2 + \left(2 \cdot \frac{4}{20} \right) = 2,63$$

$$T_{17 \rightarrow 10} = 0,01 \cdot 7 + 2 + \left(2 \cdot \frac{2}{20} \right) = 0,07 + 2 + 0,2 = 2,27$$

$$T_{10 \rightarrow 0} = 0,01 \cdot 10 = 0,1$$

Cambio di mercato = 5

$$T_0 \rightarrow 120 = 0,01 \cdot 120 + 2 + \left(2 \cdot \frac{3}{20} \right) = 1,2 + 2 + 0,3 = 3,5$$

$$T_{120 \rightarrow 130} = 0,01 \cdot 10 + 2 + \left(2 \cdot \frac{3}{20} \right) = 0,1 + 2 + 0,3 = 2,4$$

$$T_{130 \rightarrow 155} = 0,01 \cdot 25 + 2 + \left(2 \cdot \frac{5}{20} \right) = 0,25 + 2 + 0,5 = 2,30$$

SI SUPPOGA CHE NELLA CODA DELLE RICHIESTE DI UN'UNITÀ DI DISCO COMPOSTA DA 200 TRACCE, SI TROANO, NELL'ORDINE, LE RICHIESTE DEI DATI AI SEGUENTI BLOCCHI

39700 - 304 - 115 - 2600 - 2120 - 270 - 321 - 0 - 760 - 20000

LA TESTINA HA ESEGUITO L'ULTIMO MOVIMENTO PORTANDOSI DALLA TRACCIA 35 ALLA TRACCIA 97.

SI IPOTIZZI CHE LO SPOSTAMENTO DA UNA TRACCIA A UN'ALTRA RICHIEDA IN TEMPO MEDIO PARI A 40ms PER TRACCIA,

SE L'INVERSIONE DELLA DIREZIONE DI MOVIMENTO DELLA TESTINA RICHIEDA 80ms

E LA VELOCITÀ DI ROTAZIONE SIA 7200 rpm

SI VOGLIE DETERMINARE IL TEMPO RICHIESTO COMPLESSIVAMENTE, PER ACCEDERE ALLE TRACCE INDICATE PER LE POLITICHE SSTF, C-SCAN E LOOK

ER PRIMO, CALCOLO LA LATENZA ROTAZIONALE, cioé' IL TEMPO NECESSARIO PER ACCEDERE AL RECORD SELEZIONATO SUCA TRACCIA.

$$\frac{60}{2 \times 7200} = 4,17 \text{ ms}$$

POI TRASFORMO I BLOCCHI DATI IN TRACCE FACENDO i% 200

100 - 104 - 115 - 0 - 120 - 70 - 121 - 200 - 160 - 0

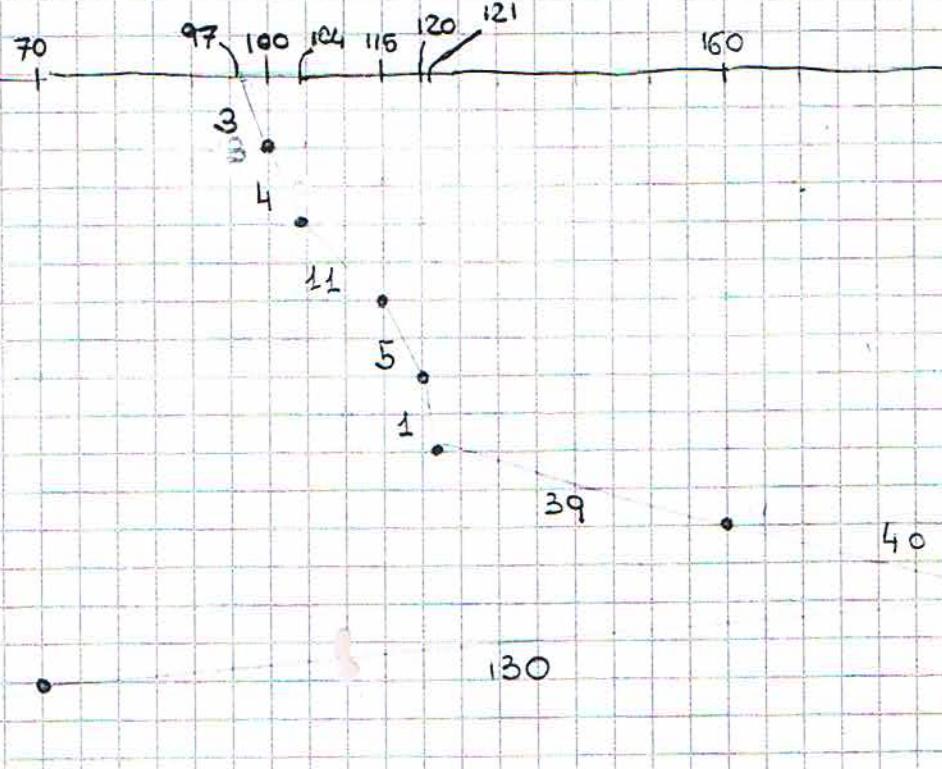
~~100~~

~~104~~

~~115~~

~~0~~

APPLICO DA SSTF QUAE SELEZIONA L'OPERAZIONE CON IL PIÙ BREVE TEMPO DI RICERCA RISPETTO ALLA POSIZIONE CORRENTE DELLA TESTINA.



IL TEMPO DI ACCESSO = (SOMMA DELLE DISTANZE \times TEMPO DI SPOSTAMENTO DI 1 TRACCIA) + INVERSIONE ROTIMENTO TESTINA + (n° SPOSTAMENTI \times DATENZA ROTAZIONALE)

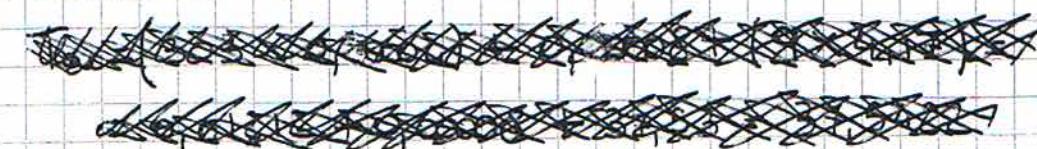
SOMMA DELLE DISTANZE = 303

TEMPO DI SPOSTAMENTO DI 1 TRACCIA = 40 ms = 0,04 ms

INVERSIONE ROTIMENTO TESTINA = 0,08 ms

n° SPOSTAMENTI = 9

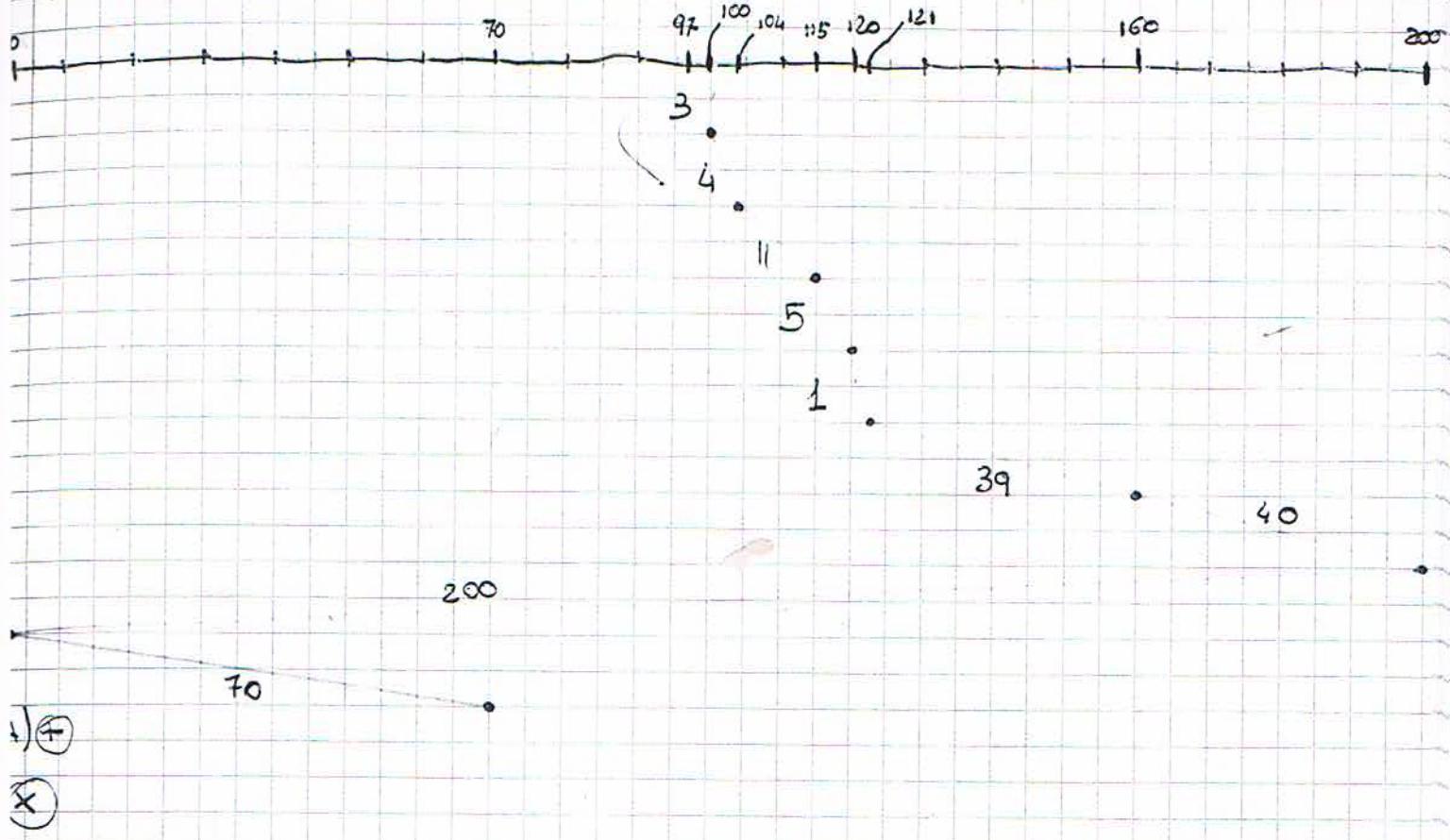
DATENZA ROTAZIONALE = 4,17 ms



TA_A = (303 \times 0,04) + 0,08 + (9 \times 4,17) = 12,12 + 0,08 + 37,53 = 49,73

16a

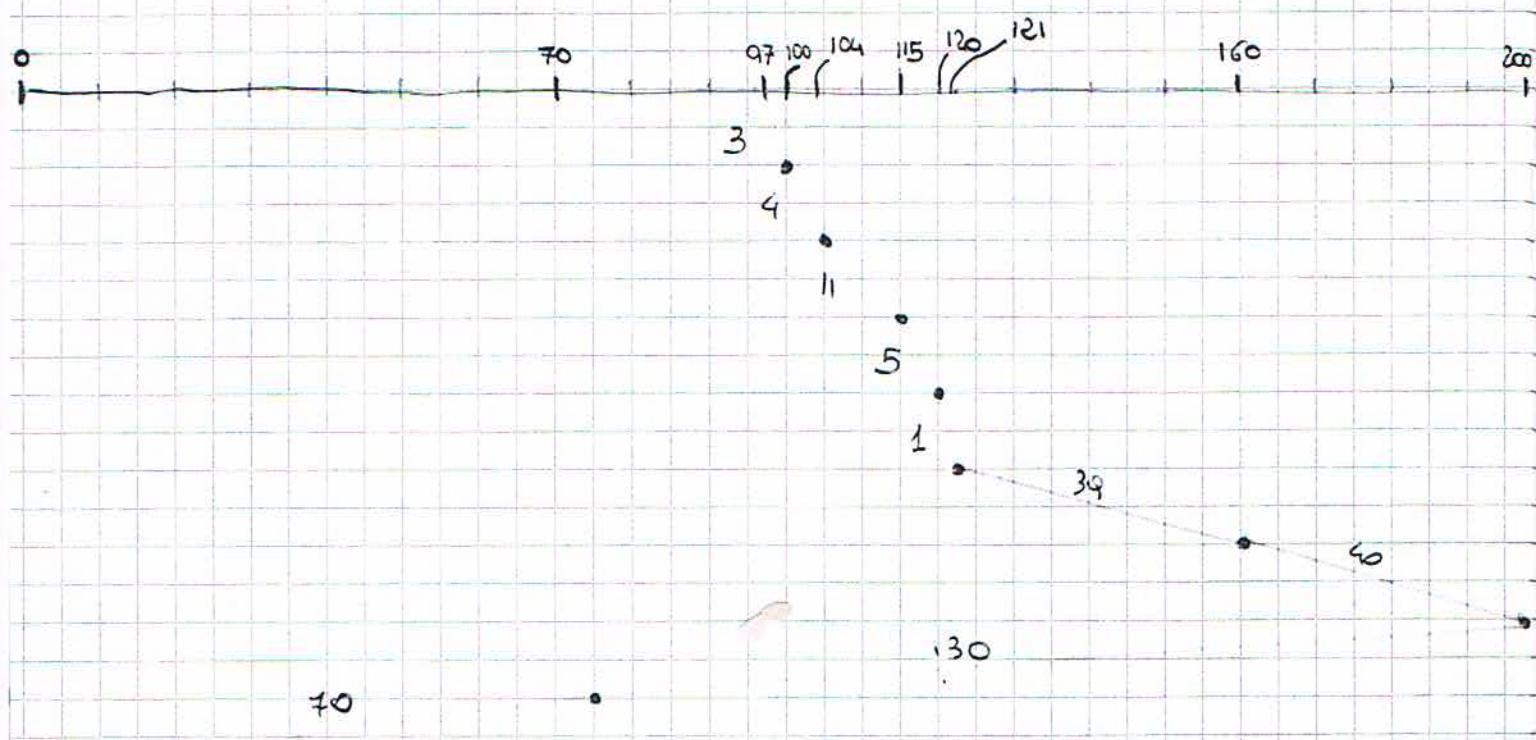
DA UN'ESTREMITÀ ALL'ALTRA DEL DISCO ED EVADE TUTTE LE
RICHIESTE CHE INCONTRA. MA SUL RITORNO NON EVADE ACCUNA
RICHiesta



$$T_{av} = (373 \times 0,04) + 0,08 + (9 \times 4,17) = 14,92 + 0,08 + 37,53 = 52,53 \text{ ms}$$

APPUNTO DA LOOK CHE PREVEDE CHE SI GUARDI SE CI SONO ALTRE RICHIESTE
Dopo l'ultima soddisfatta nella direzione di marcia corrente.

SE NON CI SONO C'È UNA DIREZIONE



CORRE DO SSTF.

TEORIA DELLA ARIA

Esercizio n° 1

Nella coda delle richieste di una unità disco composta da 200 cilindri (da 0 a 199) si trovano, nell'ordine, le richieste come (a,b) a blocchi b per ognuna delle seguenti tracce a, ognuna di 200 blocchi:

(36,1), (87,2), (56,3), (126,1), (41,4), (99,2), (164,2), (100,3), (70,2)

La testina ha eseguito l'ultimo movimento portandosi dal cilindro 89 al cilindro 90. Supponendo che l'inversione di movimento richieda 500 μ s e che lo spostamento da una traccia alla successiva richieda 30 μ s, che la latenza sia massima, si calcoli il tempo necessario per l'accesso alle tracce secondo a) FCFS, b) SSTF, c) C-SCAN.

$$m \rightarrow 30 \text{ ns} \rightarrow 0,03 \text{ ms}$$

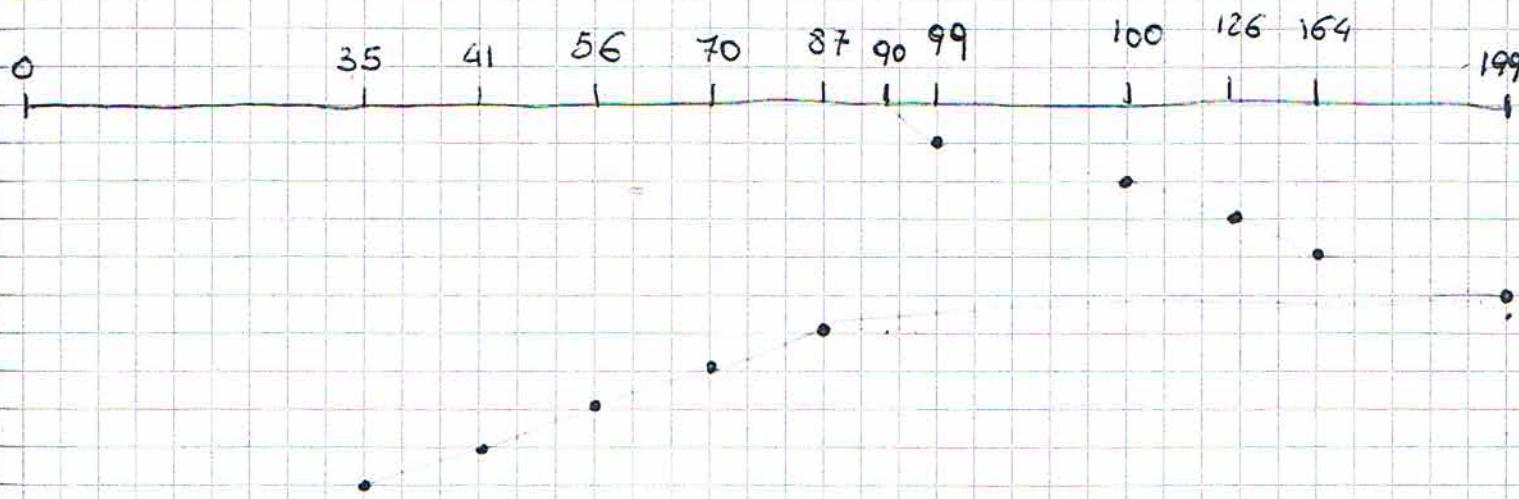
$$\text{INVERSIONE} \rightarrow 5 \text{ ms}$$

$$m \rightarrow \dots$$

$$\frac{1}{r} \rightarrow 8,34 \text{ ms}$$

$$b \rightarrow \dots$$

$$N \rightarrow 200 \quad \text{Applico da FCFS}$$



$$T_{90 \rightarrow 99} = 0,03 \cdot 9 + 8,34 + \left(8,34 \cdot \frac{1}{200} \right) = 0,27 + 8,34 + 0,0417 = 8,6517$$

$$T_{99 \rightarrow 100} = 0,03 \cdot 1 + 8,34 + \left(8,34 \cdot \frac{3}{200} \right) = 0,03 + 8,34 + 0,1251 = 8,4951$$

$$T_{100 \rightarrow 126} = 0,03 \cdot 26 + 8,34 + \left(8,34 \cdot \frac{1}{200} \right) = 0,78 + 8,34 + 0,0417 = 9,1617$$

$$T_{126 \rightarrow 164} = 0,03 \cdot 38 + 8,34 + \left(8,34 \cdot \frac{2}{200} \right) = 1,14 + 8,34 + 0,0834 = 9,5634$$

$$T_{164 \rightarrow 199} = 0,03 \cdot 35 = 2,55 \text{ ms}$$

$$\text{INVERSIONE} = 5 \text{ ms}$$

$$T_{199 \rightarrow 87} = 0,03 \cdot 112 + 8,34 + \left(8,34 \cdot \frac{2}{200} \right) = 3,36 + 8,34 + 0,0834 = 11,373$$

~~33 41 56 70 87 90 99 100 126 164 199~~

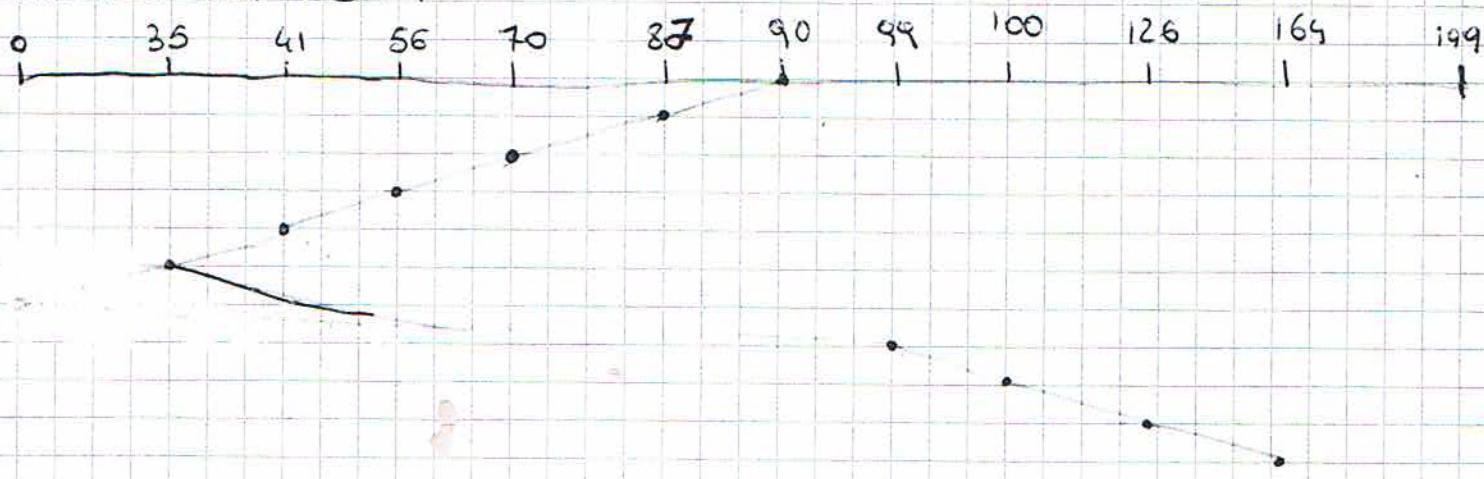
$$T_{87 \rightarrow 70} = 0,03 \cdot 17 + 8,34 + \left(8,34 \cdot \frac{2}{200} \right) = 0,51 + 8,34 + 0,0834 = 8,9334$$

$$T_{70 \rightarrow 56} = 0,03 \cdot 14 + 8,34 + \left(8,34 \cdot \frac{3}{200} \right) = 0,42 + 8,34 + 0,1251 = 8,8851$$

$$T_{56 \rightarrow 41} = 0,03 \cdot 15 + 8,34 + \left(8,34 \cdot \frac{4}{200} \right) = 0,45 + 8,34 + 0,02 = 8,81$$

$$T_{41 \rightarrow 35} = 0,03 \cdot 6 + 8,34 + \left(8,34 \cdot \frac{1}{200} \right) = 0,18 + 8,34 + 0,0417 = 8,5617$$

Applico la SSTF



INVERSIONE = 50 ms

$$T_{90 \rightarrow 87} = 0,03 \cdot 3 + 8,34 + \left(8,34 \cdot \frac{2}{200} \right) = 0,09 + 8,34 + 0,0834 = 8,5134$$

$$T_{87 \rightarrow 70} = 8,9334$$

$$T_{70 \rightarrow 56} = 8,8851$$

$$T_{56 \rightarrow 41} = 8,81$$

$$T_{41 \rightarrow 35} = 8,5617$$

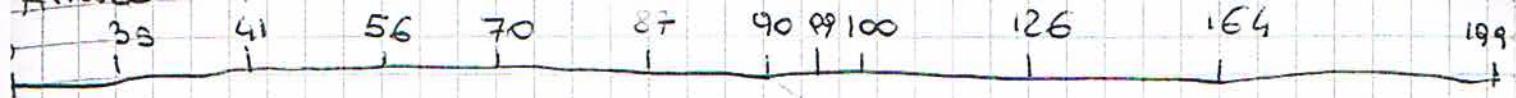
INVERSIONE = 5 ms

$$T_{35 \rightarrow 99} = 0,03 \cdot 64 + 8,34 + \left(8,34 \cdot \frac{2}{200} \right) = 1,92 + 8,34 + 0,0834 = 10,3434$$

$$T_{99 \rightarrow 100} = 8,4951$$

$$T_{100 \rightarrow 126} = 9,1617$$

$$T_{126 \rightarrow 164} = 9,5634$$



$$90 \rightarrow 99 = 8,6517$$

$$99 \rightarrow 100 = 8,1495$$

$$100 \rightarrow 126 = 9,1617$$

$$126 \rightarrow 164 = 9,5634$$

$$164 \rightarrow 199 = 0,03 \cdot 35 = 1,05$$

INVERSIONE = 5 ms

TEMPO DI ATTRaversamento = 12 ms

$$0 \rightarrow 35 = 0,03 \cdot 35 + 8,34 + \left(8,34 \cdot \frac{1}{200} \right) = 1,05 + 8,34 + 0,0417 = 9,4317$$

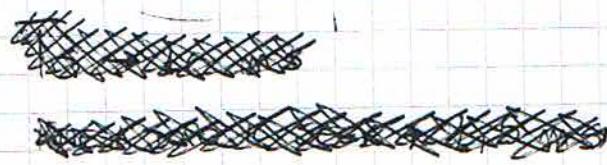
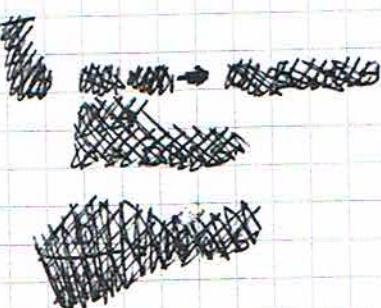
$$35 \rightarrow 41 = 8,5617$$

$$41 \rightarrow 56 = 8,81$$

$$56 \rightarrow 70 = 8,8851$$

$$70 \rightarrow 87 = 8,9334$$

Un disco ha un tempo di seek di 0.5ms per ogni cilindro attraversato, un tempo di rotazione di 3ms e un tempo di trasferimento dei dati di un settore di 10 microsecondi. Inoltre si fanno le seguenti ipotesi: la testina è inizialmente posizionata sul cilindro 98; l'attesa media prima che il settore desiderato passi sotto la testina dopo il seek è di mezza rotazione; il disco disponga di un buffer sufficiente per memorizzare un solo settore alla volta. Calcolare il tempo impiegato ad effettuare le seguenti letture: 5 settori nel cilindro 82 al tempo 0; 3 settori nel cilindro 7 al tempo 10; 9 settori nel cilindro 101 al tempo 3; 4 settori nel cilindro 56 al tempo 12; 1 settore nel cilindro 110 al tempo 8, secondo le politiche: a) SCAN, b) LOOK e c) SSTF nel caso che arrivino all'istante 13 la richiesta per la traccia 79 e all'istante 15 la richiesta per la traccia 150 sempre di due blocchi.



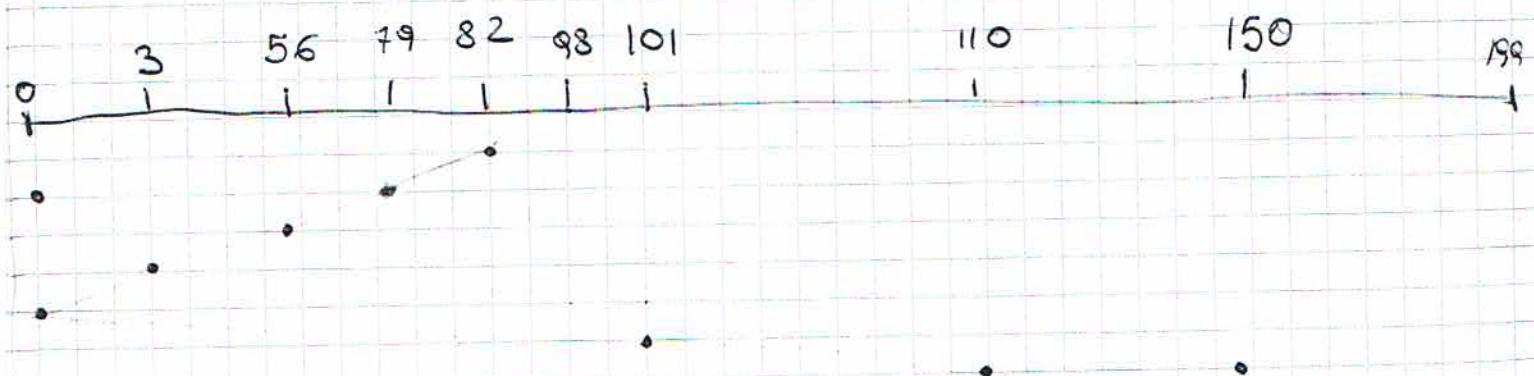
$$T_s = 0,5 \text{ ms}$$

$$T_{ATTESA} = 8,34 \text{ ms}$$

$$r = 3 \text{ ms}$$

$$\frac{b}{r} = 10 \text{ ms} \rightarrow \frac{b}{3} = 10 \text{ ms} \rightarrow b = 30 \text{ ms}$$

lettture \rightarrow 82, 3, 101, 56, 110
 ↑ ↑ ↑ ↑ ↑
 0 10 3 12 8



$$T_{98 \rightarrow 82} = 0,5 \cdot 16 + 0,3 + \left(0,3 \cdot \frac{30}{5} \right) + 8,34 = 8,3 + 1,8 + 8,34 \\ = 18,44 \text{ ms}$$

b)

ACCESSO ADDA + TCB \rightarrow 30 ms

ACCESSO A2DA + TECORIA \rightarrow 350 ms

TCB HIT RATE \rightarrow 87%

TCB MISS RATE \rightarrow 13%

PAGE FAULT RATE \rightarrow 9%

PAGE LOAD \rightarrow 20 ms \rightarrow 20 000 000 ms

DIRTY BIT OFF \rightarrow 50%

DIRTY BIT ON \rightarrow 60%

$$HIT \rightarrow 87 \times (350 + 30) : 100 = 330,6$$

In un sistema che usa paginazione, l'accesso al TLB richiede 30ns, mentre l'accesso alla memoria richiede 350ns.

a) Quando si verifica un page fault, si perdono 28ms per caricare la pagina che si sta cercando in memoria. Se il page fault rate è il 18%, il TLB hit rate è il 68%, indicare l'EAT ai dati.

b) Si supponga ora che, quando la ricerca nel TLB ha successo, la pagina riferita si trovi effettivamente in RAM, con il TLB che ha ancora un miss ratio del 13%. Si sa che il 9% degli indirizzi logici generati provoca un page fault, e nel 60% dei casi di page fault la pagina vittima ha il dirty bit a 1. Il tempo medio necessario per trasferire una pagina dalla RAM all'area di swap (o viceversa) è di 20 millisecondi. Indicare l'EAT ai dati.

a)

ACCESSO ALLA TLB \rightarrow 30 ms

ACCESSO ALLA MEMORIA \rightarrow 350 ms

TLB HIT RATE \rightarrow 68 %

~~TLB MISS RATE~~ \rightarrow 32 %

PAGE FAULT RATE \rightarrow 18 %

PAGE LOAD \rightarrow 28 ms \rightarrow 28 000 000 ns

$$\text{Hit} \rightarrow 68 \times (30 + 350) : 100 = 258,4 \text{ ms}$$

$$\text{Miss} \rightarrow [32 \times (30 + 350)] : 100 = 32 \times 730 : 100 = 233,6 \text{ ms}$$

$$\text{FAULT} \rightarrow 18 \times 32 \times 28000000 : 10000 = 1612800 \text{ ms}$$

$$\text{EAT} \rightarrow 1612800 + 233,6 + 258,4 = 1613292$$

MEMORIA TRIVALE

ESEMPIO 1

In un sistema che usa paginazione, l'accesso al TLB richiede 150ns, mentre l'accesso alla memoria richiede 400ns.

a) Quando si verifica un page fault, si perdono 8ms per caricare la pagina che si sta cercando in memoria. Se il page fault rate è il 2%, il TLB hit rate è il 70%, indicare l'EAT ai dati.

b) Si supponga ora che il sistema implementi la memoria virtuale e che, quando la ricerca nel TLB ha successo, la pagina riferita si trovi effettivamente in RAM, con il TLB che ha ancora un miss ratio del 10%. Si sa che il 4% degli indirizzi logici generati provoca un page fault, e nel 50% dei casi di page fault la pagina vittima ha il dirty bit a 0. Il tempo medio necessario per trasferire una pagina dalla RAM all'area di swap (o viceversa) è di 1 millisecondo. Indicare l'EAT ai dati.

PUNTO DI INIZIARE BISOGNA TENERE A TENTO

$$P_{R2} \times (T_{TLB} + T_{Mem}) + (P_{R2} - P_{R1}) \times (T_{TLB} + 2T_{Mem}) + (1 - P_{R1}) \times (T_{TLB} + T_{Mem} + T_{PF} + T_{TLB} + 2T_{Mem} + 8ms)$$

$$P_{R1} \times 2 \times T_{Mem} + (1 - P_{R1}) \times (T_{Mem} + T_{PF} + 2 \times T_{Mem})$$

Dove

P_{R1} → PAGE FAULT RATE

P_{R2} → TLB HIT RATE

T_{PF} → TEMPO DI PAGE FAULT

T_{Mem} → TEMPO DI ACCESSO ALLA MEMORIA

T_{TLB} → TEMPO DI ACCESSO ALLA TLB

v)

ACCESSO ALLA TLB → 150 ns

ACCESSO ALLA MEMORIA → 400 ns

TLB HIT RATE → 70%

TLB MISS RATE → 30%

PAGE FAULT RATE → 2% DEI CASI DI TLB MISS

PAGE LOAD 8 ms → 8×1000000 ns = 8000000 ns

DIRTY BIT OFF → —

DIRTY BIT ON → —

HIT → $[TLB \text{ HIT RATE} \times (\text{ACCESSO TLB} + \text{ACCESSO MEMORIA})]$: 100

$$[70 \times (150 + 400)] : 100 = 385$$

MISS → $[TLB \text{ MISS RATE} \times (\text{ACCESSO TLB} + 2 \times \text{ACCESSO MEMORIA})]$: 100

$$[30 \times (150 + 800)] : 100 = 285$$

$$\text{FAULT} \rightarrow \lfloor \text{PAGE FAULT RATE} \times \text{TLB MISS RATE} \times \text{PAGE LOAD} \rfloor \cdot 1000$$

$$= [2 \times 30 \times 8000000] : 10000 = 48000$$

EAT \rightarrow HIT + MISS + FAULT

$$385 + 285 + 48000 = 48669$$

2 TRENIT)

b)

ACCESSO ALLA TLB \rightarrow 150 ms

ACCESSO ALLA MEMORIA \rightarrow 400 ms

TLB HIT RATE \rightarrow 90%

TLB MISS RATE \rightarrow 10%

PAGE FAULT RATE \rightarrow 4%

PAGE LOAD \rightarrow 1 ms \rightarrow 1 000 000 ms

DIRTY BIT OFF \rightarrow 50%

DIRTY BIT ON \rightarrow 50%

HIT \rightarrow $\lfloor \text{TLB HIT RATE} \times (\text{ACCESSO TLB} + \text{ACCESSO MEMORIA}) \rfloor : 100$

$$\lfloor 90 \times (150 + 400) \rfloor : 100 = 495$$

MISS \rightarrow $\lfloor \text{TLB MISS RATE} \times (\text{ACCESSO TLB} + 2 \text{ ACCESSO MEMORIA}) \rfloor : 100$

$$\lfloor 10 \times (150 + 800) \rfloor : 100 = 95$$

FAULT \rightarrow $\lfloor \text{PAGE FAULT RATE} \times \text{TLB MISS RATE} \times (\text{PAGE LOAD} \times \text{DIRTY BIT OFF} + \text{PAGE LOAD} \times 2 \text{ DIRTY BIT ON}) \rfloor$

$$: 1000000 =$$

$$4 \times 10 \times (1000000 \times 50 + 1000000 \times 100) : 1000000 =$$

$$= 40 \times (50000000 + 100000000) : 1000000 =$$

$$= 6000000$$

ESERCIZIO n° 2

Si supponga che in un sistema siano presenti processi P0, P1, P2, P3, P4 ed un insieme di risorse di quattro tipi diversi A, B, C e D e di trovarsi nella seguente configurazione:

	Risorse allocate				Risorse massime				Risorse disponibili			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	6	0	Z+4	2	8	4	5	6	6	4	10	4
P1	10	0	0	2	12	7	6	8				
P2	6	2-X	0	0	8	2	0	8				
P3	0	0	3	2	0	3	4	2				
P4	4	1	2-Y	4	11	1	6	9				

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro.

CALCOLO DELLE RISORSE NEED

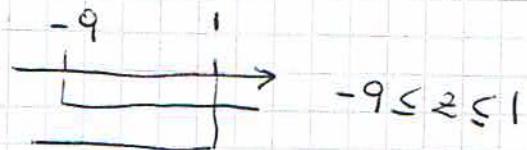
	A	B	C	D
P0	2	4	1-Z	4
P1	2	7	6	6
P2	2	X	0	8
P3	0	3	1	0
P4	7	0	Y+4	5

TRIVIATO LA SEQUENZA SICURA IN FUNZIONE DI X, Y, Z

ARTE P0

DETERMINO IL VALORE DI Z

$$\begin{cases} 1-Z \geq 0 \\ 1-Z \leq 10 \end{cases} \quad \begin{cases} -Z \geq -1 \\ -Z \leq 9 \end{cases} \quad \begin{cases} Z \leq 1 \\ Z \geq -9 \end{cases}$$



ERTINA P0
OTTENGO

$$[12 \quad 4 \quad 2+14 \quad 8]$$

ARTE P1

ERTINA P1
OTTENGO

$$[22 \quad 4 \quad 2+14 \quad 10]$$

ARTE P2

DETERMINO I VALORI DI X

$$\begin{cases} X > 0 \\ X \leq 4 \end{cases}$$

$$0 \leq X \leq 4$$

ERTINA P2
OTTENGO

$$[24 \quad X+4 \quad 2+14 \quad 18]$$

PARTE P₃
TERMINA P₃
E OTTENGO

$$[24 \quad x+4 \quad 2+17 \quad 20]$$

PARTE P₄
DETERMINO I VALORI DI Y

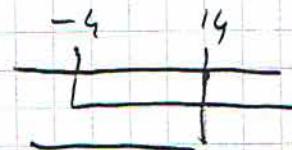
$$\begin{cases} y+4 \geq 0 \\ y+4 \leq 2+17 \end{cases}$$

$$\begin{cases} y \geq -4 \\ y \leq 2+17-4 \end{cases}$$

$$\begin{cases} y \geq -4 \\ y \leq 2+13 \end{cases}$$

PRENDO IL
MAX VALORE
DI Z

$$\begin{cases} y \geq -4 \\ y \leq 14 \end{cases}$$



$$-4 \leq y \leq 14$$

LA SEQUENZA SICURA È $\langle P_0, P_1, P_2, B_3, P_4 \rangle$ CON $0 \leq x \leq 4$,
 $y \leq 14$ e $-9 \leq z \leq 1$

Scheduling del disco

Oggetto	Valore
Cilindri	7000
Tracce per Cilindro	300
Settori per Traccia	100
Settore	1_kb
Seek Time	3 ms
Buffer di lettura	1 settore
Giri al minuto	5400
Settori letti per ogni richiesta	4
Settore in cui si trova la testina	140
Rotazione necessaria all'inizio per accedere al primo settore	1 solo giro

Richieste:

134-140-1391-477-1489-1099-1220-1075-1110-57-98

La testina si trova già sopra la traccia 140, ma comunque l'esercizio ci dice che per accedervi deve inizialmente eseguire il primo giro di disco(o piatto) mentre il seek time è uguale a zero (dato che si trova già sulla traccia 140). Ricordando che la traccia è un doppio cerchio concentrico, il cilindro è l'insieme di una certa traccia su tutti i piatti, e che i settori sono fettine all'interno della traccia.

$$\begin{aligned} \text{RPM}=5400 \Rightarrow & \quad \text{RPS} \Rightarrow 60/5400=0.011 \quad \text{RPMS} \Rightarrow 0.011 * 1000=11.11 \text{(giri per millisecondo)} \\ (\text{N.Tracce da attraversare} * \text{Seek time} + \text{T_Rotazione} + 4 * (\text{T_Rotazione}/100)) // 4 \end{aligned}$$

perché in questo esercizio i settori richiesti sono 4

SSTF(settore più vicino)

0	57	98	134	140	447	1059	1075	1110	1220	1391	1489
				X							
			X								
	X										

$$\text{Servendo } 140 = (0 + 11.11 + 4 * 0.11) = 11.54$$

$$\text{Servendo } 134 = (6 * 10 + 11.11 + 4 * 0.11) = 71.54$$

$$\text{Servendo } 098 = (36 * 10 + 11.11 + 4 * 0.11) = 371.55$$

Algoritmo del Banchiere.

Numero di processi=4

Numero di tipi di risorse=4

ALLOCATE					MASSIME				DISPONIBILI				NEED(MASSIME-ALLOCATE)				
i	X	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
p0	4	4	y-1	3	2	6	4	5	4	2	2	10	6	2	5-y	2	2
p1	8	0	z-2	2		10	1	6	8					2	1	8-z	6
p2	4	0	0	0		6	2	0	4					2	2	0	4
p3	0	1	2	2		2	3	6	2					2	2	4	0
p4	2	1	x+1	3		3	2	4	5					1	1	3-x	2

Ora verifichiamo che le 3 variabili siano comprese in intervalli validi, ovvero maggiori delle need e minori della loc.

$5-y \geq 0$ " $-y \geq -5$ " $y \leq 5$ $1 \leq y \leq 5$ (Ho usato gli apici altrimenti excel non prendeva la formula)

$y-1 \leq 0$ $y \leq 1$

$3-x \geq 0$ " $-x \geq -3$ " $x \leq 3$ " $-1 \leq x \leq 3$ "
 $x+1 \leq 0$ $x \leq -1$

$8-z \geq 0$ " $-z \geq -8$ " $z \leq 8$ " $2 \leq z \leq 8$ "
 $z-2 \leq 0$ $z \leq 2$ $z \geq 2$

Adesso cerchiamo i processi, che con il loro relativo vettore disponibilità riescono a reintrare nel vettore delle disponibilità.

Parte P0, che può essere contenuto nel vettore need.

Una volta terminata P0, il processo dealloca le sue risorse, e quindi il vettore need cresce, inglobando anche il "vettore allocate".

Quindi il vettore disp aumenta, diventando:

A B C D
6 y+1 13 8

Abbiamo assunto che y(valore compreso inizialmente tra 1 e 5), per poter essere contenuto nel vettore disponibilità abbia dovuto restringere il suo range tra 3 e 5 (altrimenti avremmo avuto valori negativi tra need[i,j]-disp[j], cosa che non deve accadere).

Quindi y ha ristretto il suo dominio tra $3 \leq y \leq 5$

Parte P1 che rientra nel nuovo vettore delle disponibilità. P1 Dealloca le sue risorse e disponibilità diventa:

A B C D
14 y+1 z+11 10

Parte P2 che rientra nel nuovo vettore delle disponibilità. P2 Dealloca le sue risorse e disponibilità diventa:

A B C D
18 y+1 z+11 10

Parte P3 che rientra nel nuovo vettore delle disponibilità. P3 Dealloca le sue risorse e disponibilità diventa:

A B C D
18 y+2 z+13 12

Parte P4 che rientra nel nuovo vettore delle disponibilità. P4 Dealloca le sue risorse e disponibilità diventa:

A B C D
20 3+y z+x+ 15

La sequenza sicura è quindi p0,p1,p2,p3,p4

ESERCIZIO N° 4

Si consideri il seguente problema:

In una fabbrica, N operai preparano piastrelle da far cuocere in un forno, capace di cuocerne M contemporaneamente. All'uscita dal forno K operai visionano le piastrelle per decorarle secondo tale sequenza di passi: se trova una piastrella difettata inizia a prenderne dal forno 2 alla volta, altrimenti ne prende 1 alla volta.

Si fornisca una soluzione usando semafori e processi. Discutere la soluzione proposta in termini di possibili situazioni di starvation/deadlock e, nel caso, si propongano soluzioni ammissibili.

nit ()

conta = 0;
semaforo operai P = N;
semaforo operai V = K;
semaforo Pin = 1;
semaforo Cott = 0;
semaforo Pout = 0;

Prodottove()

wait (Operai P); // massimo N operai
while (! FineTurno())
{
 Piastrella = Produc();
 wait (Pin); // blocca la porta del forno
 Forno [Conta] = Piastrella;
 Conta++;
 if (Conta < M)
 signal (Pin); // apre la porta del forno
 else
 signal (Cott); // cuoce piastrella
 signal (Operai P);

OtturaForno()

while (1)
{ wait (Cott); // porte entrambe chiuse
 Cuoci piastrelle();
 signal (Pout); // apre la porta
}

PARLUCK

SERVIZIO n°1

5. 3*2 = 6 punti (PER CFU 10 e 12)

Si supponga che in un sistema siano presenti processi P0, P1, P2, P3, P4 ed un insieme di risorse di quattro tipi diversi A, B, C e D e di trovarsi nella seguente configurazione:

Risorse allocate				Risorse massime				Risorse disponibili				
	A	B	C	D	A	B	C	D	A	B	C	D
P0	4	X-1	3	2	6	4	5	6	2	2	10	4
P1	8	0	Y-2	2	10	7	6	8				
P2	4	0	0	0	6	2	0	8				
P3	0	0	3	2	0	3	4	2				
P4	2	1	Z+1	4	9	1	6	9				

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro, elencando eventuali sequenze sicure e se la richiesta di P2 (2, 0, 0, 2) può essere soddisfatta.

LCALO LE RISORSE NEED = MASSIME - ALLOCATE

2	5-X	2	4	X
2	7	8-Y	6	X
2	2	0	8	X
0	3	1	0	X
7	0	5-Z	5	

SI DEVE TROVARE UNA SEQUENZA SICURA IN FUNZIONE DEI PARAMETRI X, Y, Z

E P0.

RAGIONATO IL VALORE DI X

$$\begin{cases} 5-X \geq 0 \\ 5-X \leq 2 \end{cases}$$

$$\begin{cases} -X \geq -5 \\ -X \leq -3 \end{cases}$$

$$\begin{cases} X \leq 5 \\ X \geq 3 \end{cases}$$

3 ≤ X ≤ 5

IN A P0

SI AGGIUNGE ALLE DISPONIBILI LE ALLOCATE DI P0 OTTENENDO

$$[6 \quad X+1 \quad 13 \quad 6]$$

: P3

IN A P3
SI AGGIUNGE

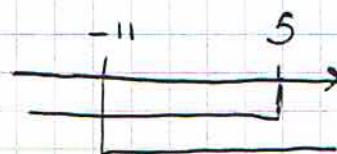
$$[6 \quad X+1 \quad 16 \quad 8]$$

R_2
RIS. P2
E OTTENGO

$$[10 \quad x+1 \quad 16 \quad 8]$$

PARTE P4
DETERMINO IL VALORE DI Z

$$\begin{cases} 5-z \geq 0 \\ 5-z \leq 16 \end{cases} \quad \begin{cases} z \leq 5 \\ z \geq -11 \end{cases}$$



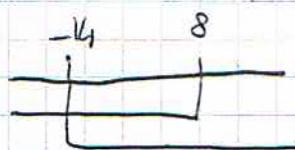
$$-11 \leq z \leq 5$$

TERMINA P4
E OTTENGO

$$[12, \quad x+2 \quad z+17 \quad 12)$$

PARTE P1
DETERMINO IL VALORE DI Y

$$\begin{cases} 8-y \geq 0 \\ 8-y \leq z+17 \end{cases} \quad \begin{cases} y \leq 8 \\ y \geq -z-9 \end{cases}$$



$$-14 \leq y \leq 8$$

PRENDO IL
VALORE MAX
DI Z

$$\begin{cases} y \leq 8 \\ y \geq -5-9 \\ y \geq -14 \end{cases} \quad \begin{cases} y \leq 8 \\ y \geq -14 \end{cases}$$

DA SEQUENZA E'

$$(P_0, P_3, P_2, P_4, P_1) \text{ CON } 3 \leq x \leq 5, \quad -14 \leq y \leq 8, \quad -11 \leq z \leq 5$$

DA RICHIESTA INIZIALE DI P2 [2 00 2] NON PUO' ESSERE IMMEDIATAMENTE SODDISFATTA PERCHE'

RISORSE ALLOCATE DI P2 + RICHIESTA DI P2 > RISORSE DISPONIBILI

$$[4 \ 0 \ 0 \ 0] + [2 \ 0 \ 0 \ 2] = [6 \ 0 \ 0 \ 2] > [2 \ 2 \ 2 \ 10 \ 4]$$

PUO' ESSERE SODDISFATTA SOLO DOPO LA TERMINAZIONE DI P0 E P3.

Esercizio n°3

5. $3 \times 2 = 6$ punti (PER CFU 10 e 12)

Si supponga che in un sistema siano presenti processi P0, P1, P2, P3, P4 ed un insieme di risorse di quattro tipi diversi A, B, C e D e di trovarsi nella seguente configurazione:

Risorse allocate	Risorse massime	Risorse disponibili
	A B C	A B C
P0	2 3+X 1	4 8 7
P1	0 3 1	3 8 3
P2	1 Y 1	3 3 2
P3	2 1 4+Z	6 3 5

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro, e se la richiesta di P0 (1, 0, 3) può essere soddisfatta.

CALCOLO LE RISORSE NEED

A B C
P0 2 5-X 6

P1 3 5 2

P2 2 3-Y 1

P3 4 2 1-Z

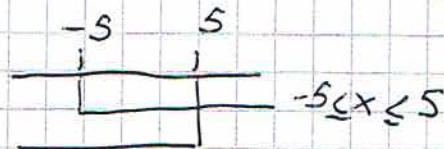
TIROVATO DA SEQUENZA SICURA IN FUNZIONE DI x, y, z

PARTE P0

DETERMINO I VALORI DI X

$$\begin{cases} 5-x \geq 0 \\ 5-x \leq 10 \end{cases}$$

$$\begin{cases} x \leq 5 \\ x \geq -5 \end{cases}$$



TERMINA P0
ED OTTENGO

$$[10 \quad x+13 \quad 11]$$

PARTE P1

TERMINA P1

ED OTTENGO

$$[10 \quad x+16 \quad 12]$$

PARTE P2

DETERMINO I VALORI DI Y

$$\begin{cases} y \geq 0 \\ y \leq x+16 \end{cases}$$

ASSURSO IL
MAX VALORE
DI X

$$\begin{cases} y \geq 0 \\ y \leq 21 \end{cases}$$

$$0 \leq y \leq 21$$

TERMINA P2
ED OTTENGO

$$[11 \quad x+y+16 \quad 13]$$

PARTE P3

DETERMINA I VACONI DI \mathbb{Z}

$$\begin{cases} 4+x \geq 0 \\ 4+x \leq 13 \end{cases} \quad \begin{cases} 2 \geq 4 \\ 2 \leq 9 \end{cases}$$

$$4 \leq x \leq 9$$

LA SEQUENZA SICURA È $\langle P_0, P_1, P_2, P_3 \rangle$ CON
 $-5 \leq x \leq 5$, $0 \leq y \leq 21$ $4 \leq z \leq 9$

DA RICHIESTA INIZIALE DI $P_0 [1, 0, 3]$ NON PUO' ESSERE IMMEDIATAMENTE
SODDISFATTA VCE'

RISORSE ALLOCATE DI P_0 + RICHIESTE DI P_0 < RISORSE DISPONIBILI

$$[2 \ 8 \ 1] + [1 \ 0 \ 3] = [3 \ 8 \ 4] < [8 \ 10 \ 10]$$

Esercizio n° 1

In un sistema con 6 frame e a partire dalla seguente tabella di informazioni:

Pagina	Tempo di caricamento	Tempo ultimo di Riferimento	Bit modifica	Bit riferimento
0	50 3	100	1	0
1	110	140	0	1
2	35 2	50	1	0
3	-	-	0	0
4	75 4	80	0	0
5	20 1	68	0	1
6	130	155	1	0
7	280	330	1	1
8	-	-	0	1
9	240	295	1	0

Determinare per la stringa di riferimento:

0 8 6 5 9 1 4 2 0 8 4 5 4 6 7 3

le pagine che vengono sostituite per gli algoritmi (a) FIFO (b) LRU e (c) CLOCK, supponendo che l'ultima pagina cambiata con questo algoritmo sia stata la pagina contenuta nel quarto frame.

ALGORITMO FIFO

PER PRIMO CARICO LE PAGINE DAL TEMPO DI CARICA DENTRO FUORI

5	5	8	8	8	8	8	8	8	8	8	8	3
2	2	2	2	2	5	5	5	5	5	5	5	5
0	0	0	0	0	9	9	9	9	9	9	9	9
4	4	4	4	4	4	4	4	2	2	2	2	2
1	1	1	1	1	1	1	1	1	0	0	0	0
6	6	6	6	6	6	6	6	6	6	6	6	4

① PAGE FAULT ② PAGE FAULT ③ PAGE FAULT ④ PAGE FAULT
 ⑤ PAGE FAULT ⑥ PAGE FAULT ⑦ PAGE FAULT ⑧ PAGE FAULT

8	8	6	6	6
5	5	5	7	7
9	9	9	9	3
2	2	2	2	2
0	0	0	0	0
4	4	4	4	4

⑤ PAGE FAULT ④ PAGE FAULT ③ PAGE FAULT

9 PAGE FAULT
SE NON ESISTE LA PAGINA
DA CARICO AL POSTO DELLA PIÙ VECCHIA
PIÙ VECCHIA (da destra)

← → INDICA DA PIÙ VECCHIA

PER PRIMO CARICO NELL'ADDESTRAMENTO MAGGIORI

7	0	8	6	5	9	1	4	2	0	8	4
9	7	0	8	6	5	9	1	4	2	0	8
6	9	7	0	8	6	5	9	1	4	2	0
1	6	9	7	0	8	6	5	9	1	4	2
0	1	6	9	7	0	8	6	5	9	1	1
4	4	1	1	9	7	0	8	6	5	9	9

⑨

PAGE
FAULT

⑥

PAGE
FAULT

⑨

PAGE
FAULT

PAGE
FAULT

PAGE
FAULT

④

⑧

⑤

①

④

②

⑩

③

5	4	6	7	3
4	5	4	6	7
8	8	5	4	6
0	0	8	5	4
2	2	0	8	5
1	1	2	0	8

PAGE
FAULT

④

PAGE
FAULT

PAGE
FAULT

11 PAGE FAULT

CARICO E AGGIORNATO SEMPRE IN CIRCA
E SPOSTATO OGNI CICLO DI DUE POSTO
POSTO

3 SEGO CON * QUESTE PAGINE CON BIT DI RIFERIMENTO 1
4 SEGO CON ← DA PAGINA VITIMA

3	*
4	
5	*
6	
7	*
8	*

SE LA PAGINA VITIMA HA D'ASTERISCO

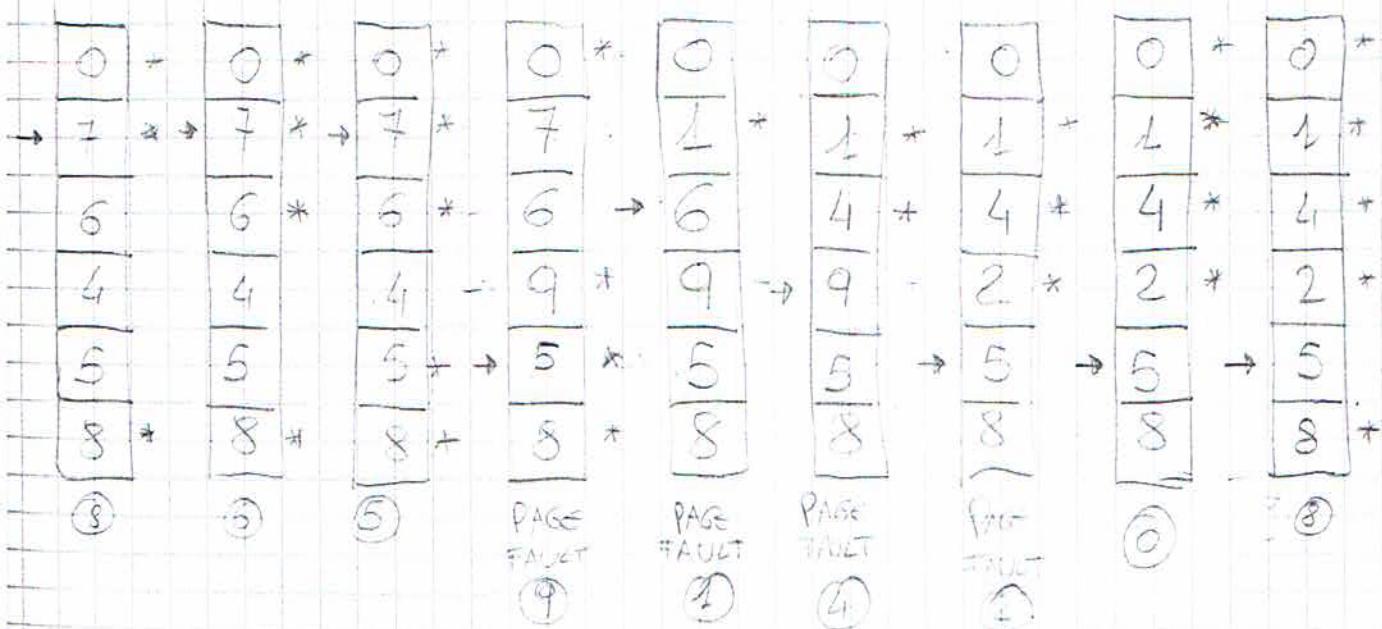
TOLGO GLI ASTERISCI ATUTTI PARTENDO DALLA PAGINA VITIMA FINO A QUANDO TROVO UNA PAGINA SENZA ASTERISCO

DETTO D'ASTERISCO E SOSTITUISCO IL VALORE DI PAGINA

DETTO DA → ALLA PAGINA SUCCESSIVA A QUESTA SOSTITUITA

SE CARICO UNA PAGINA PRESENTE DETTO *

PAGE
FAULT
⑥



0	*	0	*	0	*	0	*	0	*	3	*
1	*	1	*	1	*	1	*	1	*	1	*
4	*	4	*	4	*	4	*	4	*	4	*
2	*	2	*	2	*	2	*	2	*	2	*
5	*	5	*	5	*	5	*	6	*	6	*
8	*	8	*	8	*	8	*	7	*	7	*

④ ⑤ ④

PAGE
FAULT

⑥

PAGE
FAULT

⑦

PAGE
FAULT

⑧

8 PAGE FAULT

In un sistema con 6 frame e a partire dalla seguente tabella di pagine:

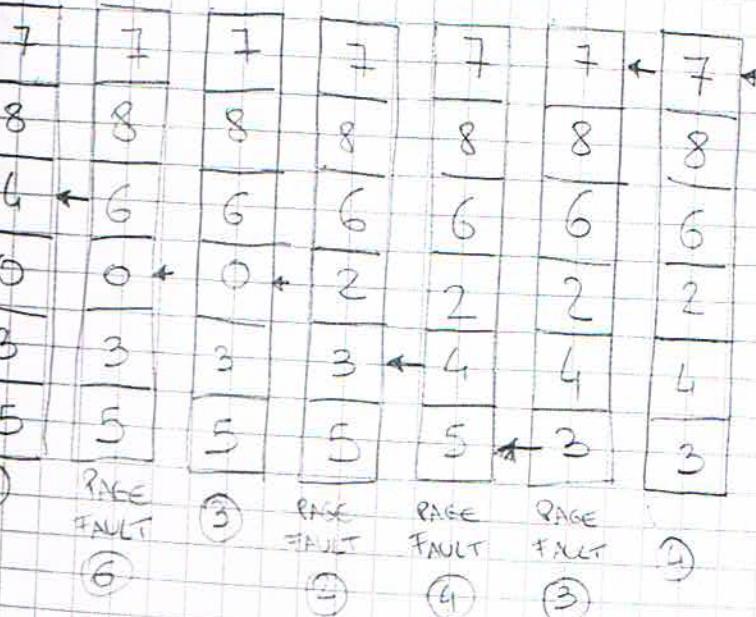
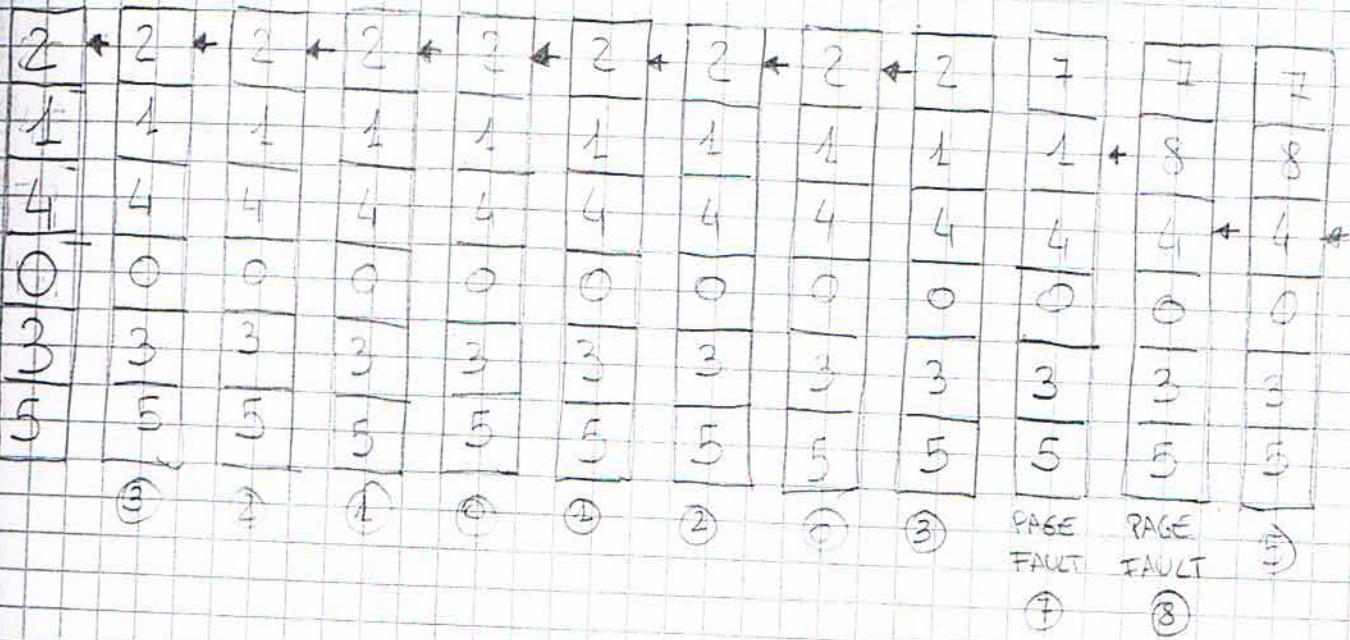
Pagina	Tempo di caricamento	Tempo ultimo di Riferimento	Bit modifica	Bit riferimento
0	104	260	0	0
1	-	-	0	0
2	20	270	1	0
3	-	-	0	1
4	-	-	0	0
5	302	303	1	0
6	-	-	0	1
7	45	305	1	1
8	203	260	0	1
9	79	250	0	0

Determinare per la stringa di riferimento:

(3) 2 1 0 1 2 0 3 0 8 5 7 6 3 2 4 3 4

le pagine che vengono sostituite per gli algoritmi (a) FIFO (b) LRU e (c) CLOCK, supponendo che l'ultima pagina cambiata con questo algoritmo sia stata la 4 e le tabelle di pagine relative.

ALGORITMO FIFO



6 PAGE FAULT

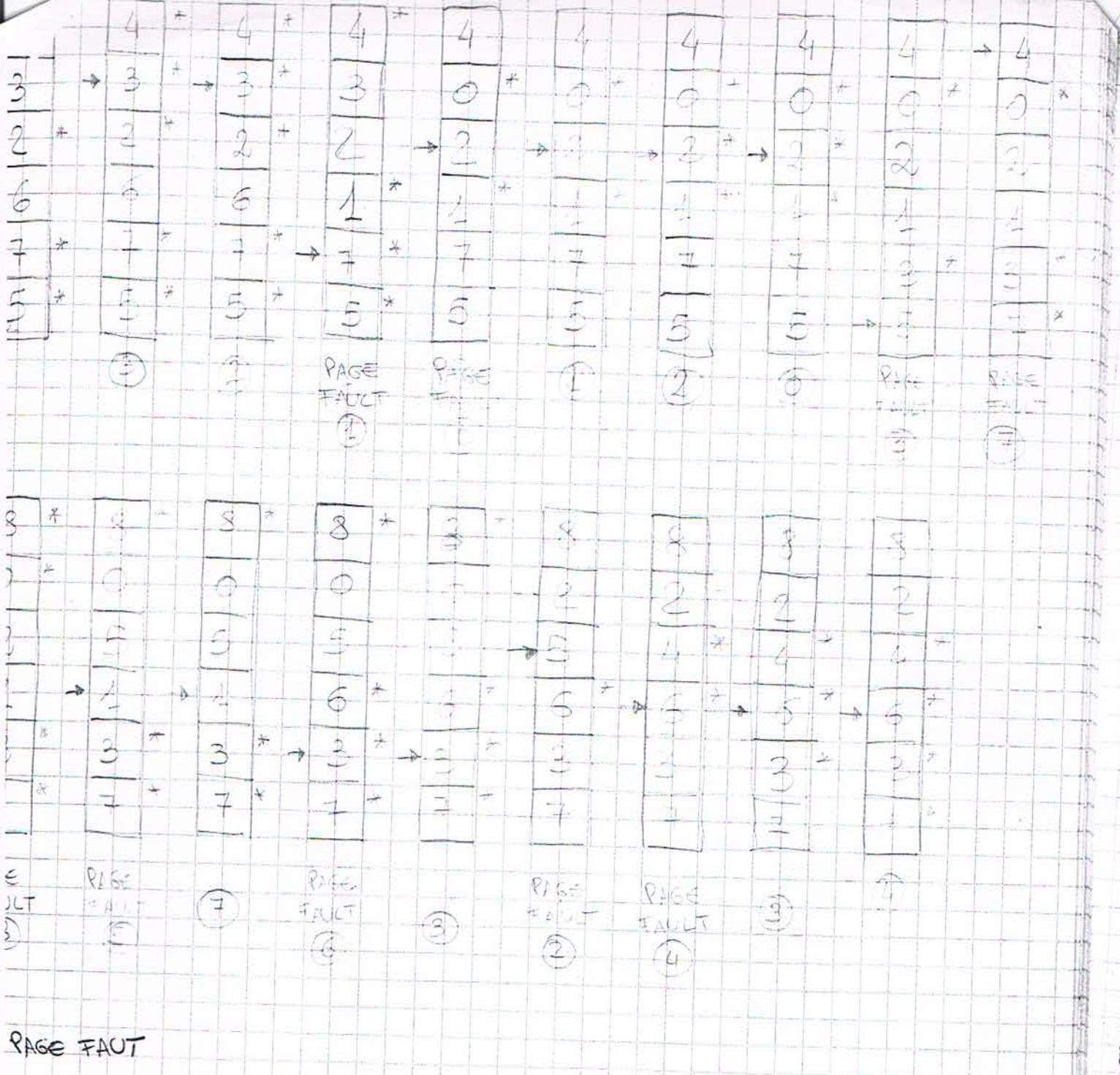
ALGORITMO LRU

1	3	2	1	0	1	3	0	3	7	8
5	1	3	2	1	0	4	1	0	3	7
2	5	1	3	2	2	0	4	2	0	3
0	2	5	5	3	3	3	2	1	2	0
3	0	0	0	5	5	5	5	1	2	0
4	4	4	4	4	4	4	4	4	3	1
(3)	(2)	(4)	(0)	(1)	(2)	(0)	(3)	PAGE FAULT	PAGE FAULT	PAGE FAULT
								(7)	(8)	(5)

7	6	3	2	4	3	6
5	7	6	3	2	4	3
8	5	7	6	3	2	2
2	8	5	7	6	6	6
0	3	8	5	2	7	7
2	0	0	8	5	5	5

6 PAGE FAULT

(7) PAGE FAULT (3) PAGE FAULT (9) PAGE FAULT
 (6) (2) (4) (7) (7)



PAGE FAULT

FILE SYSTEM

Esercizio n°1

Si consideri un file system che utilizza 16 blocchi fisici ed usa la tecnica della bitmap per tenere traccia della memoria libera. Si consideri la seguente configurazione iniziale della bitmap :

1000 1100 0000 1010

con la convenzione che nella bitmap lo stato del blocco 0 è rappresentato dal bit più a sinistra e 1 rappresenta i blocchi occupati. Supponendo che il file system allochi sempre i blocchi di indice minore, descrivere la struttura della bitmap dopo le seguenti operazioni:

- a) scrittura del file A (4 blocchi); b) scrittura del file B (3 blocchi); c) aggiunta di due blocchi ad A cancellazione di B

di Buccu

ALLOCAZIONE CONTIGUA

CONFIGURAZIONE INIZIALE

1000 1100 0000 1010

SCRITTURA DEL FILE A (4 blocchi)

1000 1111 1100 1010

SCRITTURA DEL FILE B (3 blocchi)

1111 1111 1100 1010

AGGIUNTA DI DUE BLOCCHI AD A

1111 1111 1111 1010

CANCELLAZIONE DI 2 BLOCCHI B

1100 1111 1111 1010

ALLOCAZIONE CONCATENATA

CONFIGURAZIONE INIZIALE

1000 1100 0000 1010

SCRITTURA DEL FILE A (4 blocchi)

1111 1110 0000 1010

SCRITTURA DEL FILE B (3 blocchi)

1111 1111 1100 1010

AGGIUNTA DI DUE BLOCCHI AD A

1111 1111 1111 1010

CANCELLAZIONE DI 2 BLOCCHI DI B

1111 1111 0000 1010

ALLOCAZIONE INDICIZZATA

CONFIGURAZIONE INIZIALE

1000 1100 0000 1010

SCRITTURA DEL FILE A (4 blocchi)

1111 1111 0000 1010

SCRITTURA DEL FILE B (3 blocchi)

1111 1111 1111 1010

AGGIUNTA DI DUE BLOCCHI AD A

1111 1111 1111 1111

CANCELLAZIONE DI 2 BLOCCHI DI 3

1111 1111 1111 1000